

# NETWORK ATTACKING USING DRONE

\*S.Likhitesh, #T.S.R. Maanvi, \*A.Mohan

<sup>\*,#</sup>UG Student, <sup>\$</sup>Asst.Professor, CBIT Hyderabad and India.

<sup>\*</sup>slikhit@cbit.ac.in, <sup>#</sup>ruchieatha97@gmail.com, <sup>\$</sup>mohanmtech9@gmail.com

Abstract - The insecure nature of networks poses a grave threat to the underlying IT infrastructure of any organization. The present work aims to create a drone that tests and attacks networks within its range, reporting the same in the form of saved data files. It can be controlled through a networked connection. The drone shall have the flexibility to be run by any Linux powered microcontroller, with the necessary interfaces (communication and control) built. Such drones can be made commercially available and shall help automate the internal security auditing of an organization's critical IT infrastructure. The drone shall test the nearby networks against a wide variety of standard network attacks, allowing the flexibility to perform custom designed attacks on the fly. The work presented here is, an attempt to unify two unrelated domains - automation and robotics; with network security. The expected outcome is a drone that can be remotely controlled, with the capability to do everything in a networked environment that any standard linux machine can do.

Keywords : Network attack, media access control(Mac), ARP.

# I. INTRODUCTION

Aerial vehicles, especially unmanned aerial vehicles (UAV) are emerging as a new technology domain, finding many uses in almost every sector of life. In addition to the traditional UAV's which were once RC piloted, todays technological advances focus mainly on autonomous behavior and mini atomization of such vehicles.

Securing critical network infrastructure has never been such an important issue as it is today. However, a mechanism is not available till date to remotely audit network infrastructure for weakness or vulnerability. Responding to the emerging trend of personal UAVs, the current project aims to explore the uses of such unmanned aerial vehicles in the field of network security, and how they can be leveraged to ease up the life of pen tester or a network administrator.

This work aims to deal with the lack of an efficient automatable platform for testing network infrastructure. Also, to test the network infrastructure spread over larger landscapes, this project aims to provide a remote solution in the form of a drone that carries with it a full Linux operating system, and can be controlled remotely. The drone thus becomes a remotely controlled aviary computer which has the capabilities of testing networks against insecure connections.

To create a remotely operable aerial reconnaissance vehicle that can remotely be piloted, and carries with it a full Linux operating system that has the capabilities of testing network infrastructure. The Linux box shall be responsible to handle all the motion control of the drone, and provides a remote interface to control the network testing toolkit that it houses.

Currently there exist network auditing tools that are too rigid in their implementation, many of which are not cross platform and are also not generic. Several existing tools are also not open sourced. Added to everything, they require the testing personnel to move around with the entire machine in hand to be able to test for vulnerabilities in a network that is spread over a larger geographical area, for example, colleges and corporate buildings.

There are aerial vehicles which serve the purpose of consumer product delivery or agricultural monitoring systems. These happen to be autonomous, but lack in the capability of being aware of the networking environment around them, making them susceptible to remote takeover.

An aerial vehicle, which was chosen to be a quadcopter, which would carry with it a complete Linux operating system. The operating system shall be equipped with network analysis frameworks that can check for vulnerabilities specified by the auditor. They can also be scripted and run to automate the process. Additionally, there shall be the capability to delegate heavy processing intensive applications to the operating base; So that a record of the audit can be maintained even in the unfortunate event of unrecoverable damage to the drone.

## Advantages of proposed system:

• It provides the capability to remotely survey a networked environment.



• Corporates and organizations can use such devices to speed up their network auditing mechanisms.

# **II. LITERATURE SURVEY**

# 2.1 Quadcopter

A quadcopter is a type of an unmanned aerial reconnaissance vehicle, which runs on the basic principles of aerodynamics. Not delving deep into the physics involved, every quadcopter is expected to have the following basic components:

- 1. Four propellers, two clockwise and two counter clockwise. Attached to four motors, these form the propulsion system of the quadcopter.
- 2. A flight controller that controls the power delivered to each of the motors to position the drone for a stable flight
- 3. A frame, which is light enough, so the thrust produced by propellers can lift it.
- 4. A battery pack with sufficient discharge rate to power the propulsion system [21].

## Applications

- Quadcopters are a useful tool for university researchers to test and evaluate new ideas in a number of different fields, including flight control theory, navigation, real time systems, and robotics. There are numerous advantages to using quadcopters as versatile test platforms. They are relatively cheap, available in a variety of sizes and their simple mechanical design means that they can be built and maintained by amateurs.
- The largest use of quadcopters in the USA has been in the field of aerial imagery. Quadcopter UAVs are suitable for this job because of their autonomous nature and huge cost savings. Drones have also been used in light-painting photography.
- In 2014 The Guardian reported that major media outlets have started to put serious effort into exploring the use of drones for reporting and verifying news on events that include floods, protests and wars. Some media outlets and newspapers are using drones to capture photography of celebrities.
- Quadcopters have also been used in various art projects including but not limited to drone photography. They may be used in performance art with new degrees of positional control that allows for new uses of puppets, characters, lights and cameras. They have also been used in light shows
- Quadcopters are used all over the world for racing (also known as "drone racing") and freestyle

events. Racing and freestyle quadcopters are built for speed and agility. Racing and freestyle drones tend to be relatively small in size, with 250 mm between the propeller shafts and/or 5-6 inch props being the usually upper end of the size scale

# 2.2 SSH

SSH, or Secure Shell, is a remote command line interface to an operating system, usually Linux and Unix based. It is very similar to the Telnet utility the only difference is that SSH uses cryptographically secure encryption standards in its communications as opposed to Telnet that uses plain text communications.

# 2.3 WebSocket

WebSocket is a bidirectional fully duplex communications protocol over a single TCP channel. It is used to communicate over the web, much like the HTTP protocol, but allows communications with lower overheads.





# . Figure 3.1. Network Attacking Drone

As shown in figure, Elaborating the data flow further, the input from the control station (which can be command or control) is received by either of the two independent servers (SSH and WebSocket) running on the Raspberry Pi. Like can be seen, The SSH server is responsible for handling the commands required for interacting with the network environment, while the web sockets server handles the commands responsible for the fight control. The SSH server can be used to invoke necessary scripts that analyze and reckon the networks or attack them. The information is sent back to the control station over SSH. The commands delegated to the flight controller by the web sockets server are executed by the flight controller, and any feedback of the sates of motors (such as failures) is sent back over WebSocket's. This information can then be used to take further steps, such as invoking the necessary scripts to perform suitable attacks.

The advantage of using two independent servers is that each can be equipped with its own internet interface, thereby allowing any final work to be accomplished in the unfortunate event of any damage to any single communication interface. Also, each module can be run independently with its own dedicated system resources, like



a dedicated processor core and memory page, thereby ensuring independent subsystems working at their peak performance [3].

# **IV. IMPLEMENTATIONS**

## 4.1 Building the Quadcopter

# **Parts Required:**

- Frame 1x (F450)
- 4x Brushless DC Motors
- 4x ESC (Electronic Speed Controllers)
- 1x LiPo Battery (12V)
- 2x Clockwise Propellers
- 2x Counter Clockwise Propellers

# Assembling the Parts:

The four arms of the Frame are connected to the power distribution board that comes inbuilt with the frame. The assembly can be done referencing a manual that comes with the frame itself. The wire leads are soldered to the outputs of the power distribution board. The motors are attached to the four arms and ESCs are attached to the motors. The ESCs take a pulse width modulated signal from the flight controller and accordingly vary the output voltage to the motors. The input voltage to the ESCs comes from the power distribution board. The ESCs are connected to this board. The clockwise and counter clock wise propellers are attached at the respective places and this completes the building of the frame.

The Arduino Nano and the MPU 6050 are connected via wire leads and the combination serves as the flight controller. This is placed in the central hollow of the drone to provide maximum stability and closeness to the center of gravity. The Raspberry Pi is placed on the top shelf of the drone, and is connected via USB to the underlying Arduino Nano. The power supply to the Flight controller is obtained via the Pi's USB terminal, and a 5V - 12A DC Power source, powers the Pi.

The power to the motors and the ESCs is delivered via the LiPO battery which is taped to the bottom of the drone, and this assembly completes the building of the drone.

## 4.2 Working of the Quadcopter

The Raspberry Pi is the brain of the device. It runs a web sockets server which receives connection from the web sockets client, which, in this case happens to be an Android application. This communication channel is used to transmit three-dimensional motion commands to the drone. These commands are parsed and delegated to the underlying flight controller[21].

It also runs an SSH server that provides a remote access to the operating system interface. This channel is used to test, attack and survey the networks around by making use of the operating system capabilities. SSH service is used to remotely access the capabilities of the Linux operating system and invoke the necessary scripts for network reconnaissance/surveillance or attacks.

The combination of an Arduino Nano and IMU MPU 6050 serves as the flight controller.

The MPU 6050 uses its inbuilt accelerometer and gyroscope to calculate the three-dimensional state of the drone, and passes the information to Arduino Nano. The Arduino Nano board then takes this information, receives the motion commands from the Pi, calculates the amount of thrust needed at each motor to shift the vehicle from its current state (given by MPU 6050) to the required state (given by the Pi). Appropriate signals are then sent to each of the four electronic speed controllers that accordingly vary the speed of the motor they are connected to, thereby changing the thrust supplied by each motor[3].

# **Protocols Used**

# SSH

SSH service is used to remotely access the capabilities of the Linux operating system and invoke the necessary scripts for network reconnaissance/surveillance or attacks.

# WebSocket's

WebSocket protocol is used to send navigation commands to the drone via an Android application that functions as the control station

# Networking Scripts

# **Description of concepts**

The network attacking scripts have been divided into two broad categories. Structures, where the definitions of standard communication protocol packets have been defined and implemented. Attacks, which use the structures available in the Structures part and customize the behavior and flow of these packets. Brief description of each follows [20].

## Structures

The structures have been classified and categorized according to where they belong to in the OSI architecture. Each protocol implementation is available in its corresponding layer.

## Layer 2

Layer 2 is the immediate layer above the physical layer, and in the OSI model, it deals with managing raw bit addresses and managing communication interfaces. It has a variety of protocols like Ethernet, Ethernet II, ARP, RARP etc. For this project, Ethernet and ARP have been implemented at the Layer 2 level.



#### Ethernet

The original Ethernet (IEEE 802.3) is a protocol designed to negotiate addressing at the physical level between two communicating end points. It's main objective is to uniquely identify a network interface card and provide for a communication facility between them.

The Ethernet protocol has three fields in it (excluding the preamble).

A packet that implements the Ethernet protocol is appended to any payload that needs to be transmitted. Any end point that receives this packet can then check for the field destination MAC address. If it matches with that of its own, it accepts the packet and forwards it to the kernel. Or else it simply discards it.

#### ARP

The Address Resolution Protocol is designed to link logical IP addresses at the higher level with the physical MAC addresses at the lower hardware level. At the beginning of communication, every device only has its physical MAC address. A service like DHCP that runs on the network may then assign them logical addresses. A logical address belongs to a communication end point. It is not a property of the device located at that point. For instance, if two devices at two end points are swapped, their IP address are swapped too. This is because the IP address did not belong to the systems, rather to the end points to which the systems were connected.

If two devices in a network wish to communicate with each other, the only information available to them is the logical address of the corresponding end point. There has to be a way to identify which hardware address is now sitting at the end point. Because the behavior of network interface cards is such that if the Ethernet header destination MAC address corresponds to it's own MAC address, it accepts the packet. Thus, ARP is used to bind hardware MAC addresses to logical addresses. A logical address can either be IPv4 or IPv6 (in the future) ARP proceeds as follows.

Every device maintains a cache table of hardware address – logical address association entries. This is called the ARP cache. Initially this cache is empty at each device. When a device wishes to communicate with another device whose logical address it knows, it broadcasts an ARP request asking for the MAC address of the corresponding logical address, and inserts its own hardware address – logical address association entry in the packet. If a device is present at the requested end point, it updates its ARP cache with the association entry of the sender and then sends an ARP reply to the sender, informing it of its own hardware address. The original sender then updates its ARP cache with the association entry, and a communication may now proceed. ARP also has the facility to send broadcast messages about updates in a device's association entry. If a device with a static logical address joins a network and wishes to declare its association entry, it sends a gratuitous ARP message to the broadcast informing all devices of its hardware MAC address with IP address and so on. Port scanning is usually done by attempting to make a full TCP connection to all the TCP ports. Open ports respond with a successful establishment of connection. The aim of a port scan is to determine the services running over a target device. For example, open port 80 indicates a webserver is running. This information can then be used to fingerprint the operating system. A suitable exploit can then be found that remotely exploits the given combination of operating system and services. Or else, attempts may be made to have access remotely via default credentials. For eg, default SSH credentials, or default FTP server credentials etc.

## **Packet Injections**

Packet Injections are a type of attack where stray packets are injected into the network with the intention of disrupting normal communication. When rogue packets are injected, they can cause unexpected behavior, which can range from crashing services to flooding a network, or leaking credentials to arbitrary code execution.

Two types of packet injections have been implemented in the current project [1].

# ARP Poisoni<mark>ng</mark>

ARP poisoning is the use of gratuitous ARP packets to intentionally modify the ARP cache of devices in a network. The defacto implementation is to poison the cache such that attacking node appears as the router to the client and as client to the router.

# Engi Procedure

- 1. Send gratuitous ARP to client to update gateway IP with attacker MAC
- 2. Send gratuitous ARP to gateway to update client IP with attacker MAC
- 3. Enable IP forwarding in the attacker machine

In some cases, it might be needed to poison all the devices in a network. In such a case, the gratuitous ARP packets are sent as addressed to broadcast, and no packets are sent to the gateway. This however makes the attacker unable to capture communication packets from the gateway to the client.

## **Deauth Frames**

This is an attack where WLAN De-authentication Frames are injected into the network. Either targeted to a specific device or addressed to broadcast, thereby de-authenticating all stations. For targeted de-authentication, packets are also sent to the AP. Thus the station is de-authenticated from the



AP and the AP is de-authenticated from the station[4],[5].

# V. RESULT ANALYSIS

#### 4.1 Establishing a connection



#### Figure 4.1Connection Establishment

**Description:** Figure 4.1 shows the connection request from the android app to the WebSocket server running on the raspberry pi. The android application acts as the WebSocket client and this application is used to control the motion of the drone

# 4.2 Acquiring network configurations

The first logical step to do is to identify what network the drone is in, so we employ sniffing for this purpose.



Figure 4.2 Execution for Sniffing the network

Sec. 1						
				managed.pcap		60
Fig. D	dit View Co Co	oture Analyze Statist	ics Telephony wireless 1	cools Help		
				and Med.		
A #	1 2 3 🖕	xou(	2 * H H 🖵 🖂	이민리표		
	ly a display filter					Expression
No	Time	Course	Destination	Protocol Leosth is	de.	
	1.0.0000000	102 168 2 1	210 255 255 255	Dicition Configuration	OTTEX A HITE/1 4	
	2.0.000000	192,168,2,1	239,255,255,258	.550P 355 N	OTTEX * HTTP/1.1	
	3.0.000000	192,168,2,1	239,255,255,250	SS2P 300 N	OTTEY * HTTP/1.1	
	4 9,000000	192.168.2.1	239,255,255,258	SSDP 353 N	OTIFY * HTTP/1.1	
	5.0.000000	192,168,2,1	239,255,255,259	SSDP 355 N	OTIFY * HTTP/1.1	
	6 9,000000	192,168,2,1	239.255.255.259	SSDP 365 N	OTIFY * HTTP/1.1	
	7 8.000998	192.168.2.1	239.255.255.258	SSDP 359 N	OTIFY * HTTP/1.1	
	8 0.000998	192.168.2.1	239.255.255.250	SSDP 300 N	OTIFY * HTTP/1.1	
	9.0.000080	192.168.2.1	239.255.255.250	SSDP 371 N	OTIFY * HTTP/1.1	
4.						
0000	01 00 5e 7f ff	fa 94 44 52 07 m	3 89 08 00 45 00	D RE.		
0010	01 5f 34 7a 00	\$ 00 04 11 ce 70 c	as 02 01 of ff	12p		
00/29	ff fa 67 6c 61	60 81 40 df fe 4	1 47 54 49 46 59	1.1.KNOTIFY		
000,000	20 28 20 40 04	1 04 00 21 31 20 3	1 80 88 48 67 73	HTTP/ 1.1. MOS		
	38 38 31 39 30	A 38 8d 8a de 54 3	a 75 72 AP 3a 73 Bit	1900 NT:urn:s		
0050						
0050	63 68 65 6d 61	1 73 2d 77 69 66 6	3 61 6c 6c 69 61 che	mas-w ifiallia		
0058 0050 0070	63 68 65 6d 63 6e 63 65 2d 6f	1 73 2d 77 69 66 6 72 67 3a 73 65 7	9 61 6c 6c 69 61 che 2 76 69 63 65 38 nce	Has-w ifiallia t-org: service:		
0058 0058 0070 0050	63 68 65 6d 6; 6e 63 65 2d 64 57 46 41 57 40	1 73 2d 77 69 66 6 f 72 67 3a 73 65 7 1 41 4e 43 6f 6e 6	9 61 6c 6c 69 61 che 7 76 69 63 65 38 nce 1 69 67 3a 31 6d WFA	mas-w ifiallia H-org: service: GMLANC onfig:1.		
0050 0050 0050 0050 0050	63 68 65 6d 6 6e 63 65 2d 6 57 46 41 57 4c 0a 4e 54 53 34	1 73 2d 77 69 66 6 f 72 67 3a 73 65 7 ; 41 4e 43 6f 6e 6 i 73 73 64 70 3a 6	9 61 6c 6c 69 61 che 2 76 69 63 65 3a nce 3 69 67 3a 31 6d WF4 1 6c 69 76 65 6d .N1	Has-w ifiallia -org: service: GAANC onfig:1. 'S:ssd p:alive.		
0050 0070 0050 0050 0050	63 68 65 6d 6 6e 63 65 2d 6 57 46 41 57 40 9a 4e 54 53 34 9a 4c 6f 63 61	1 73 2d 77 69 66 6 7 72 67 3a 73 65 7 3 41 4e 43 6f 6e 6 1 73 73 64 70 3a 6 1 74 69 6f 6e 3a 6	9 61 60 60 69 61 che 2 76 69 63 65 3a not 6 69 67 3a 31 6d WFA 1 6c 69 76 65 6d .NT 1 74 74 70 3a 2f .Lc	mas-w ifiallia -org: service: MLANC enfig:1. /Sissd pialive. catio n:http:/		
	63 68 65 6d 6: 6e 63 65 2d 69 57 46 41 57 4d 9a 4e 54 53 3d 9b 4c 6f 63 61 2f 31 39 32 24	1 73 2d 77 69 66 6 f 72 67 3a 73 65 7 ; 41 4e 43 6f 6e 6 1 73 73 64 70 3a 6 1 74 69 6f 6e 3a 6 1 31 36 30 2e 32 2	9 61 6c 6c 09 61 chr 2 76 69 63 65 3a nc 6 69 67 3a 31 6d WFA 1 6c 69 76 65 6d .NN 1 74 74 70 3a 2f .Lc 1 31 3a 38 30 2f /10	mas-w ifiallia -org: service: WLANC onfig:1. Sissd pialive. icatio n:http:/ 2.168 .2.160/		
0050 0050 0050 0050 0050 0050 0050 005	63 68 65 6d 6: 6e 63 65 2d 6: 57 46 41 57 4c 0a 4e 54 53 3d 0a 4c 6f 63 61 2f 31 39 32 2e 09 67 64 2e 78 64 3a 03 20	1 73 2d 77 69 66 6 f 72 67 3a 73 65 7 ; 41 4e 43 6f 6e 6 1 73 73 65 7 1 73 73 64 70 3a 6 1 74 69 6f 6e 3a 6 1 31 36 38 2e 32 2 1 6d 6c 6d 0a 55 5 1 70 78	9 61 6c 6c 09 61 che 2 76 69 63 65 3a nce 6 69 67 3a 31 0d wFA 6 69 67 3a 31 0d wFA 1 6c 69 76 65 0d .N 1 74 74 70 3a 2f .Lc 1 31 3a 38 30 2f /11 5 4c 3a 75 75 69 100 5 70 70 30 27 4 4	mas-w ifialla -org: service: MLANC enfig:1. Sissd pialive. icatio n:http:// 2.168.2.1:80/ S.xml. USN:uui icooco m.cooc.		
0050 00270 0050 0050 0050 0050 0050 0050	63 68 65 6d 6: 6e 63 65 2d 61 57 46 41 57 4c 0a 4e 54 53 3d 0a 4c 6f 63 61 2f 31 39 32 2e 69 67 64 2e 75 64 3a 30 36 36 39 30 31 2	1 73 2d 77 69 66 6 7 72 67 3a 73 65 7 41 6e 43 6f 6e 3a 1 73 73 64 70 3a 6 1 74 99 6f 6e 3a 6 1 31 36 3a 2e 32 2 1 6d 6c 0d 0a 55 5 1 30 30 30 30 30 2d 3 31 3a 3a 30 30 2d 3	9 61 6c 6c 69 61 che 2 76 69 63 65 3a ncc 6 69 67 3a 31 6d WFA 1 6c 69 76 65 6d .NT 3 74 74 70 3a 27 LC 1 13 3a 38 30 27 /11 1 4c 3a 75 75 69 1gc 1 3c 30 30 36 3c 2d d: 1 3c 34 34 35 37 ent	<pre>mas-w ifialls borg: service: bMLANC emfig:1. Sissd pialive. catio mihttp:// 2.168 2.2.180/ 5.xml. USN:uui 00000 00-0000-000-0000</pre>		
00505 00500 00500 00500 00500 00505 00505 00505 00505 00505 00505	63 68 65 6d 6: 6e 63 65 2d 61 57 46 41 57 4c 0a 4e 54 53 3d 0a 4c 6f 63 01 2f 31 39 32 2e 69 67 64 2e 76 64 3a 30 36 36 30 30 31 2c 38 37 61 33 32	1 73 2d 77 69 66 6 7 72 67 3a 73 65 7 7 41 4e 43 66 6e 6 1 73 73 64 70 3a 6 1 74 30 6f 6e 3a 0 9 31 36 30 2e 32 2 5 6d 6c 0d 0a 55 5 1 30 30 30 30 2d 3 1 31 30 3a 3a 75 72 6	9 61 6c 6c 69 61 che 2 76 69 63 65 3a 1 6c 69 73 a 31 6d wFA 1 6c 69 76 65 6d .N 7 74 74 70 3a 2f .L 1 3a 38 30 2f .1 1 4e 3a 75 75 69 19 1 4e 3a 73 75 69 10 1 34 34 34 35 32 00 1 34 34 36 66 65 077	rms-w ifiallia + org: service: MAANC enfig:1. 5:ssd p:alive. (2:168.2.1.80/ 5:ml. USN:UU1 00000 00-600- 1:100 0-344452 3851: wrn.scbe		
00505 00502 00500 00500 00500 00500 00500 00500 00500 00500 00500 00500 00500 00500 00500	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	1         73         2d         77         69         66         6           7         2         67         3a         73         65         7.           0         41         4a         36         66         66         67         73         64         70         3a         6           1         73         54         70         3a         6         66         60         1         73         73         64         70         3a         6         1         73         56         66         6         66         6         70         3a         53 <td>9 51 6C 5C 09 61 cht 2 76 69 63 65 3a 11 6d 8 69 67 3a 31 6d wfA 1 6C 69 70 65 6d .N1 9 74 74 70 3a 27 .16 9 31 3a 38 30 27 /11 3 4c 3a 75 75 69 10 1 3 3a 38 32 20 97 1 3 3a 36 66 65 077 1 3 3a 36 66 65 077 1 3 63 66 65 077 1 3 66 61 60 65 ma</td> <td>rms-w_lfiallia iorg:service: bLANC enfig:1. 5:ssd pialive. icatio nihttp:// 2:168 .2.1:80/ 1:xnl. USR:uui 00000 00-06600- 1:100 0-944452 1389:: urn:sche -wifi alliance</td> <td></td> <td></td>	9 51 6C 5C 09 61 cht 2 76 69 63 65 3a 11 6d 8 69 67 3a 31 6d wfA 1 6C 69 70 65 6d .N1 9 74 74 70 3a 27 .16 9 31 3a 38 30 27 /11 3 4c 3a 75 75 69 10 1 3 3a 38 32 20 97 1 3 3a 36 66 65 077 1 3 3a 36 66 65 077 1 3 63 66 65 077 1 3 66 61 60 65 ma	rms-w_lfiallia iorg:service: bLANC enfig:1. 5:ssd pialive. icatio nihttp:// 2:168 .2.1:80/ 1:xnl. USR:uui 00000 00-06600- 1:100 0-944452 1389:: urn:sche -wifi alliance		

Figure 4.3. Available Network Configurations

**Description:** The above figures show one way of obtaining the network configuration state by using sniffing (promiscuous mode). These network configurations are acquired by executing commands on the drone remotely via SSH connection.

#### 4.3 Passive Attacks

Passive attacks do not involve in any interaction with the network around them. They stealthily recon for information. Sniffing is the best example, used in almost all passive attacks[4].

#### **Promiscuous Mode Sniffing**



Figure 4.3(a) Server and Client Execution

ans 🖨	wireshark *				Mon 201				T
					manag	ed.pcap			
Ele j	dit Yiew Go G	apture Analyze	Statistics Tele	phony Wire	iess Iools Help				
41	1 0 0	0 8 8 0	2 ( ) 2	h4 -		12			
-		121 121		<u> </u>		444		-	
App	ity a display filter							- E	xpression.
No.	Time	Source	D	estination	Protocol	Length info			
	1 0.086000	192.158.2	1 2	19.255.255	250 550P	* VALTON 696	HTTP/1.1		
	2 0.005009	192.168.2	.1 2	19.255.255	.250 SSDP	355 MOTIFY *	HTTP/1.1		
	3 0,008000	192.168.2	.1 2	19.255.255.	.250 SSDP	300 NOTIFY *	HTTP/1.1		
	40.000000	192.168.2	.1 2	39.255.255.	.250 SSDP	353 NOTIFY *	HTTP/1.1		
	5 0,000000	192,168,2	.1 2	19.255.255.	,250 SSDP	355 NOTIFY *	HTTP/1.1		
	5 8,00000	192.168.2	.1 2	39.255.255.	.250 SSDP	365 NOTIFY *	HTTP/1,1		
	7 0,000000	192.168.2	.1 2	39.255.255.	.250 SSDP	359 NOTIFY *	HTTP/1.1		
	8 0.000000	192.168.2	.1 2	39.255.255.	.250 SSOP	388 NOTIFY *	HTTP/1.1		
	9.000000	192,168,2	.1 2	19.255.255.	250 SSOP	371 NOTIFY *	WTTP/1.1		
<ul> <li>France</li> <li>Ethic</li> <li>Inti-</li> <li>User</li> </ul>	me 1: 305 byte: ernet II, Src: ernet Protocol r Datagram Prot	s on wire (25 BelkinIn_07: Version 4, 5 locol, Src Po	20 bits), 365 a3:89 (94:44: irc: 192.168.2 irt: 1900, Dst	bytes cap 52:07:a3:8 .1, Dst: 2 Port: 190	tured (2920 bits 9), Dst: IPv4mca 39.255.255.250 0	) st_7f:ff:fa (01:	00:5e:7f:ff:fa)		
<ul> <li>Frame</li> <li>Ethe</li> <li>Inter</li> <li>Use</li> <li>Sim</li> </ul>	me 1: 305 bytes ernet II, Src: ernet Protocol r Datagram Prot ple Service Din	s on wire (25 BelkinIn_07: Version 4, S tocol, Src Po LCOVERY Proto	28 bits), 365 a3:89 (94:44: irc: 192.168.2 irt: 1900, Dst col	bytes cap 52:07:a3:8 .1, Dst: 2 Port: 190	tured (2920 bits 9), Dst: IPvdmca 39.255.255.250 0	) st_7f:ff:fa (01:	00:5e:7f:ff:fa)		
<ul> <li>Fram</li> <li>Ethe</li> <li>Ints</li> <li>Use</li> <li>Sim</li> <li>0000</li> </ul>	me 1: 365 bytes ernet II, Src: ernet Protocol r Datagram Prot ple Service Din 01 00 5e 7f f	s on wire (25 BelkinIn_07: Version 4, 5 tocol, Src Po scovery Proto	20 bits), 365 a3:89 (94:44: rrc: 192.168.2 rrt: 1900, Dst col 52 07 a3 89 00	bytes cap 52:07:a3:8 .1, Dst: 2 Port: 190	tured (2929 bits 9), Dst: IPvdmca 29.255.255.259 0	) st_7f:ff:fm (01: .E.	00:5e:7f:ff:fa)		
<ul> <li>Fram</li> <li>Ethic</li> <li>Ints</li> <li>Use</li> <li>Sim</li> <li>0000</li> <li>0010</li> </ul>	me 1: 305 bytes ernet II, Src: ernet Protocol r Datagram Prot ple Service Din 81 00 5e 7f f 81 5f 34 7m 0	s on wire (25 BelkinIn_07: Version 4, 5 tocol, Src Po scovery Proto f fa 94 44 1 0 00 04 11 0	120 bits), 385 a3:89 (94:44: irc: 192.168.2 irc: 1900, Dst icol 52 07 a3 89 00 ce 78 c0 a8 02	bytes cap 52:07:a3:87 .1, Dst: 2 Port: 190 00 45 00 01 0f ff	tured (2929 bits 9), Dst: IPvdmca 39.255.255.259 0 ^0 R	) st_7f:ff:fm (01: .E.	00:5e:7f:ff:fa)		
<ul> <li>Fram</li> <li>Ethic</li> <li>Interview</li> <li>User</li> <li>Sim</li> <li>0000</li> <li>0010</li> <li>0020</li> </ul>	me 1: 305 bytes ernet II, Src: ernet Protocol r Datagram Proto ple Service Din 01:00 5e 7f f 01 5f 34 7m 0 ff fa 07 6c 0	s on wire (25 BelkinIn_07: Version 4, 5 tocol, Src Po scovery Proto f fa 94 44 1 0 00 04 11 0 7 6c 01 4b 0	120 bits), 365 #3:89 (94:44: inc: 192.168.2 inc: 1900, Dst incol 52 07 #3 89 00 ce 78 c8 #8 02 df fe 4e 4f 54	bytes cap 52:07:a3:8 1, Dst: 2 Port: 190 0 045 00 01 ef ff 49 46 59	tured (2920 bits 9), Dst: IPv4mca 29.255.255.250 0 	) st_7f:ff:fa (01: .E. JFY	00:5e:7f:ff:fa)		
<ul> <li>Fram</li> <li>Ethic</li> <li>Ints</li> <li>Use</li> <li>51m</li> <li>0000</li> <li>0010</li> <li>0020</li> <li>0020</li> </ul>	me 1: 305 bytes ernet II, Src: ernet Protocol r Datagram Proto ple Service Din 81 00 5e 7f f 81 5f 34 7m 0 ff fm 07 6c 0 20 2m 20 48 5	s on wire (25 BelkinIn_07: Version 4, 5 tocol, Src Po scovery Proto f fa 94 44 1 0 60 04 11 0 7 6c 01 4b 1 4 54 50 2f 1	128 bits), 365 a3:89 (94:44: irc: 192.168.2 irc: 1900, 0st ircol 52 07 a3 89 00 ce 70 c0 a8 00 df fe 4e 4f 54 31 2e 31 00 00	bytes cap 52:07:a3:8 (1, Dst: 2) Port: 190 00 45 00 01 ef ff 49 46 59 148 6f 73	tured (2920 bits 9), Dst: IPvdmca 29.255.255.250 	) st_7f:ff:fa (01: .E. .E. Hos	00:5e:7f:ff:fa)		
<pre>&gt; Fram &gt; Eth &gt; Ints &gt; Use &gt; 51m 00000 0010 0020 0020 0020</pre>	me 1: 305 bytes ernet II, Src: ernet Protocol r Datagram Prot ple Service Dir 01:00 5c 7f f 01:5f 34 7m 0 ff fa 07 6c 0 20 2a 20 48 5 74 3a 32 33 3	s on wire (25 BelkinIn 07: Version 4, S tocol, Src Po scovery Proto 7 fa 94 44 1 0 00 04 11 1 7 6c 01 4b 1 4 54 50 2f 3 9 2e 32 35	128 bits), 365 a3:89 (94:44: irc: 192.188.2 rt: 1900, Dst col 52 07 a3 89 00 co 78 c0 a8 02 df fe 4e 4f 54 31 26 31 00 00 15 2e 32 35 33	bytes cap 52:07:a3:80 1, Dst: 2 Port: 1900 01 ef ff 49 46 59 48 6f 73 2e 32 35	tured (2920 bits 9), Dst: IPvdmca 89.255.255.250 0 	) st_7f:ff:fa (01: .E. .F. IFY Hos .25	00:5e:7f:ff:fa)		
<pre>&gt; Fram &gt; Ethy &gt; Ints &gt; Use &gt; 51m 0000 0010 0020 0020 0020 0020 0020 00</pre>	me 1: 305 byter ernet II, Src: ernet Protocol r Datagram Prot ple Service Dir G1 00 5c 7f f G1 5f 34 7m 0 ff fm 07 6c 0 G2 02 20 48 5 74 3m 32 33 3 30 3m 31 90 3	s on wire (25 Belkinin 07: Version 4, 5 tocol, Src Po tocol, Src Po toco	128 bits), 365 a3:89 (94:44: irc: 192.168.2 irc: 1900, Dst irci 52 07 a3 89 00 co 78 co 88 00 df fe 4e 4f 54 31 2e 31 04 00 35 2e 32 35 35 4e 54 3a 75 77 4e 46 54 54 35 2e 32 35 35 4e 54 3a 75 77 4e 46 54 54 55 26 32 35 35 56 55 55 57 55 58 55 59 55 50 50 50 55 50 55 50 50	bytes cap 52:07:a3:80 1, Dst: 2 Port: 1900 01 ef ff 49 46 59 48 6f 73 2e 32 35 6e 38 73	tured (2920 bits 9), Dst: IPvdnca 20 255.255.250 	) st_7f:ff:fa (01: .E. .E. .E. .E. .E. .25 n:s	00:5e:7f:ff:fa)		
<pre>&gt; Frm &gt; Eth &gt; Int: &gt; Use: &gt; 51m 0000 0010 0020 0030 0030 0050 0050 0020</pre>	me 1: 365 bytes ernet II, Src: ernet Protocol ple Service Dir 01 00 5e 7f f 01 5f 34 7a 0 ff fa 07 6c 0 20 2a 29 48 5 74 3a 32 33 3 30 3a 31 39 3 63 68 65 66 5	s on wire (25 BelkinIn 07: Version 4, 5 tocol, Src Po beovery Probo f fa 94 44 1 0 00 04 11 4 54 50 2f 9 2e 32 35 0 30 0d 0a 1 73 2d 77 6 22 67 25	128 bits), 365 a3:89 (94:44: rrc: 192.108.2 trc: 192.108.2 trc: 1900, Dist col 52 07 a3 89 00 cc 78 cc 9 a8 02 df fe 4e 4f 54 31 2e 31 00 00 f5 2e 32 35 33 4e 54 3a 75 72 69 66 69 61 66	bytes cap 52:07:a3:8 1, Dst: 2 Port: 190 0 45 00 01 ef ff 49 46 59 48 6f 73 52 32 35 66 3a 73 66 06 61 63 05 3a	tured (2920 bits 9), Dst: IPv4nce 29.255.255.250 	) st_7f:ff:fa (01: .E.  IFY Hos .25 n:S ISA Cal	00:5e:7f:ff;fa)		
<pre>&gt; Frm &gt; Eth &gt; Int: &gt; Use: &gt; 0000 0010 0020 0020 0020 0020 0020 0020</pre>	me 1: 305 byte: ernet II, Src: ernet Protocol r Datagram Prot gle Service Dir 0: 00 5c 7f f 0: 5f 34 7a 0 0: 20 2a 20 46 5 74 3a 32 33 30 3a 31 30 30 36 65 66 6 57 46 45 74 57 46 74 57 57 76 57 57 76 57 57 76 57 57 76 57 57 76 57 57 76	s on wire (25 Belkinlm 07: Version 4, 5 tocol, Src PL tocol, Src PL toco	228 bits), 365 a3:80 (94:44: irc: 192.108.2 irc: 192.108.2	bytes cap 52:07:a3:8 1, Dst: 2 Port: 190 00 45 00 01 ef ff 49 46 59 48 6f 73 20 32 35 66 3a 73 60 65 3a 3a 31 ed	tured (2320 bits 9), Dst: IPvince 9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	) .E. .E. .FY Hos .25 n15 11 ce: -1	00:5e:7f:ff(fa)		
> Fram > Eth > Int: > Use: > Sim 0000 0010 0020 0020 0020 0020 0020 002	me 1: 305 byte: ernet II, Src: rnet Protocol r Datagram Prot ple Service Dir 61 60 5e 7f f 61 5f 34 7a 0 ff fa 07 6c 0 20 2a 26 45 74 3a 32 33 3 30 3a 31 39 3 30 38 65 46 6 57 46 45 57 4 6a 45 53 4 57 46 45 57 4	s on wire (25 Belkinle,07: Version 4, S tocol, Src Pc bovery Probo f fa 94 44 1 0 00 04 11 0 7 6c 01 4b 4 54 50 2f 1 9 2e 32 35 0 30 0d 0a 1 73 2d 77 1 f 72 67 3a c 41 4e 43 c 47 37 64	128 bits), 365 a3:89 (94:44: rfc: 192.108.2 rft: 1900, Dat col 52 07 a3 89 00 cc 70 co 88 00 df fc 4e 4f 54 32 co 31 ad 00 35 2c 32 35 33 4e 54 3a 75 73 65 66 69 61 66 73 65 72 76 61 67 6a 61 6e 68	bytes cap 52:67:33:8 1, Dst: 2 Port: 100 00 45 60 01 ef ff 49 46 59 48 6f 73 2c 32 35 5 6e 3a 73 6 c 69 61 63 65 3a 3a 31 6d 76 65 6d	tured (2320 bits 9), Dst: IP-Mae 9), Dst: IP-Mae 9 	) st_7f:ff:fa (01: .E.  IFY Nos .25 nis ce: .11 .12 .25 .25 .25 .25 .25 .25 .25 .25 .25 .2	98:5e:7f:ff:fa)		
> Fram > Eth > Int: > Use: > Sim 0000 0010 0020 0020 0020 0020 0000 000	me 1: 305 byte: ernet II, Src: rnet Protocol r Datagram Proto ple Service Dir 61 00 5e 7f f 61 5f 34 7a 0 20 2a 26 45 74 3a 32 33 30 3a 31 39 3 63 68 66 8d 6 57 46 41 57 4 6a 4c 54 53 3 6a 4c 6f 63 6	s on wire (25 Belkinln 07: Version 4, 5 tocol, Src Pc beovery Proto 7 fa 94 44 1 6 00 04 11 7 6c 01 4b 17 6c 01 4b 17 6c 01 4b 1 73 2d 77 f 72 67 3a c 41 4e 43 c 41 4e 43 a 73 73 64 1 74 69 6f	128 bits), 365         a3:89 (94:44:         irc: 192.108.2         irc: 192.108.2         irc: 192.08.2         irc: 192.08.2         irc: 192.08.2         irc: 192.08.2         irc: 192.108.2         irc: 192.08.2         irc: 192.108.2         irc: 192.108	bytes cap 52:07:33:8 1, DBE: 2 Port: 100 01 ef ff 49 46 59 48 6f 73 2e 32 35 6e 3a 73 6e 3a 73 6e 6 49 61 63 65 3a 3a 31 6d 76 65 6d	tured (2320 bits 9), Dst: IP+Mcs 9), Dst: IP+Mcs 9), Dst: IP+Mcs 9 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,	) st_7f:ff:fa (01: .E.	00:5e:7f:ff:fa)		
<ul> <li>Fram</li> <li>Ethio</li> <li>Ints</li> <li>Uset</li> <li>S100</li> <li>0010</li> <li>0020</li> <li>00200</li> <li>0020</li> <li>00200</li> <li>002</li></ul>	me 1: 305 bytte: ernet II, Src: ernet Protocol r Datagram Proto ple Service Di 61 60 5c 7f f 61 5f 34 7a 0 ff fa 07 6c 0 20 2a 26 45 5 74 3a 32 33 3 30 3a 31 99 30 68 65 68 6 6c 63 65 2d 6 6c 63 65 2d 6 6a 46 54 53 3 6a 4c 6f 63 6 2f 31 33 22 2f 31 33 32	s on wire (25 Belkinln 07: Version 4, 1 tocol, Sre Pe tovery Proto f fa 94 44 1 9 00 04 11 1 7 6c 01 4b 1 4 54 50 2f 1 9 2e 32 35 0 30 0d 0a 4 1 73 2d 77 1 f 72 67 3a a 73 73 64 1 74 69 6f 1 e 31 36 38 1	128 bits), 365         a3:89 (94:44:         rc: 192.108.2         rrt: 1900, Dst         col         52 07 A3 89 00         66 69 61 61 60         73 65 72 76 61         60 63 61 60 70         70 3a 61 6c 65         66 83 81 60 74         70 23 22 83 23         60 36 87 47         72 32 28 31 33	bytes cap 52:07:33:8 1, Dst: 2 Port: 100 01 ef ff 49 46 59 48 6f 73 2e 32 35 6e 3a 73 6e 69 61 63 65 3a 3a 31 6d 76 65 8d 79 3a 2f	tured (2320 bits) 9), Dst: IP-Mac 9), Dst: IP-Mac 9, 255.255.259 9 	.E. .F. .F. .F. .FY HOS .25 .15 .15 .15 .15 .15 .15 .15 .15 .15 .1	98:5e:7f:ff:fa)		
<ul> <li>Fram</li> <li>Ethio</li> <li>Ints</li> <li>Uset</li> <li>S100</li> <li>0010</li> <li>0020</li> <li>00200</li> <li>002000</li> <li>002000</li> <li>00200&lt;</li></ul>	me 1: 305 bytes ernet II, Src: ernet Protocol r Datagram Proto pls Service Dir 61 00 5c 7f f fa 07 5c 0 61 5f 34 7a 0 ff fa 07 5c 0 20 2a 26 45 5 74 3a 32 33 3 30 38 61 56 45 6 6c 63 65 26 6 6r 45 57 4 6a 45 57 4 6a 46 56 3 8a 4c 6f 63 6 2f 31 39 32 2 96 67 64 2c 7	s on wire (25 BelkinIm,07: Version 4, 2 toccol, Src Pr becovery Prote f fn 94 44 1 0 60 04 110 7 6c 01 4b 4 54 50 27 9 20 32 55 0 30 00 06 a 1 73 24 77 7 17 24 77 3a c 11 46 2 1 73 73 46 4 1 74 69 6f e 31 36 38 1 56 46 c 6d	120       bits), 365         a3:80       (94:44:         rf:       192.168.2         rf:       1900, Dst         icol       0         52       07 a3 89         60       0         52       07 a3 89         61       16         52       07 a3 89         65       23 1 84         64       47 55         65       22 35         34e       54 6 49         65       62 68         66       69 61         66       68 69         67       3a 61 6c         66       63 a6         67       3a 68         67       3a 64         68       63         77       3a         68       63         67       3a         68       64         70       3a         70       3a         86       63         97       3a         98       48         98       48         98       48         98       48         98       48 <t< td=""><td>bytes cap 52:67:43:8 (1, Dst: 2: Port: 100 60 45 60 61 ef ff 49 46 59 48 6f 73 2e 32 35 6e 3a 73 6c 69 61 63 65 3a 3a 31 06 76 36 45 76 36 45 76 55 66 77 38 47 83 38 22 76 55 66</td><td>Lured (2329 bits a), Dit: TPvdnca 29.255.256.259 </td><td>) st, 7f:ff:fa (01: .E.  IFY Moss       </td><td>98:5#:7f:ff:fa)</td><td></td><td></td></t<>	bytes cap 52:67:43:8 (1, Dst: 2: Port: 100 60 45 60 61 ef ff 49 46 59 48 6f 73 2e 32 35 6e 3a 73 6c 69 61 63 65 3a 3a 31 06 76 36 45 76 36 45 76 55 66 77 38 47 83 38 22 76 55 66	Lured (2329 bits a), Dit: TPvdnca 29.255.256.259 	) st, 7f:ff:fa (01: .E.  IFY Moss       	98:5#:7f:ff:fa)		
<ul> <li>Fran</li> <li>Eth</li> <li>Int:</li> <li>Use:</li> <li>Sim</li> <li>0000</li> <li>0010</li> <li>0020</li> <li>00200</li> <li>0020</li> <li>00200</li> <li>0020</li> <li>0020</li> <li>0020</li> <li< td=""><td>me 1: 305 bytes ernet IT, Src: ernet Protocal r Datagram Protocal Die Service Di Die Service Di Die Service Di Co 2a 26 45 5 74 3a 32 33 3 03 3a 31 30 3 63 68 65 64 6 66 63 65 26 6 68 45 7 46 45 57 4 68 45 64 57 4 68 45 64 53 6 27 11 39 32 2 59 67 64 22 7 44 3a 38 38 3</td><td>s on wire (25 BelkinIm, 67: Version 4, 1 toccol, Src Pr becovery Prots f fm 94 44 1 toccol, Src Pr 96 60 44 11 7 6c 01 45 4 54 50 27 1 9 26 32 35 0 30 00 0m 1 73 20 27 1 7 74 69 6f 1 75 6f 1 74 69 6f 1 74 60 6</td><td>228 bits), 305 a3:80 (94:44: rrt: 19400, Dist tocol 52 07 a3 09 0 52 07 a3 09 0 for 4e 4f 55 13 2e 31 04 0 15 2e 32 55 2 25 35 2e 32 55 32 4e 54 3a 75 7; 99 66 69 61 66 76 66 66 69 6 67 6a 61 6c 66 66 66 69 6 67 6a 36 07 47 2e 32 2e 31 38 38 55 53 4 33 39 38 24 39 32</td><td>bytes cap 52:07:03:8: 1, Det: 2 1, Det: 2 1907: 1900 01:07 49:45:50 49:45:5</td><td>tured (2320 bits ), Dit: JPv4nca 9), Dit: JPv4nca 9), Dit: JPv4nca 9, 255.255.256 9 1, 1, 2, 1, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,</td><td>) st_7f:ff:fa (01: .E.  JFY Hos .25 n:s lia ce: .1. ve. .pi/ 00-</td><td>00:50:7f;ff;fa)</td><td></td><td></td></li<></ul>	me 1: 305 bytes ernet IT, Src: ernet Protocal r Datagram Protocal Die Service Di Die Service Di Die Service Di Co 2a 26 45 5 74 3a 32 33 3 03 3a 31 30 3 63 68 65 64 6 66 63 65 26 6 68 45 7 46 45 57 4 68 45 64 57 4 68 45 64 53 6 27 11 39 32 2 59 67 64 22 7 44 3a 38 38 3	s on wire (25 BelkinIm, 67: Version 4, 1 toccol, Src Pr becovery Prots f fm 94 44 1 toccol, Src Pr 96 60 44 11 7 6c 01 45 4 54 50 27 1 9 26 32 35 0 30 00 0m 1 73 20 27 1 7 74 69 6f 1 75 6f 1 74 69 6f 1 74 60 6	228 bits), 305 a3:80 (94:44: rrt: 19400, Dist tocol 52 07 a3 09 0 52 07 a3 09 0 for 4e 4f 55 13 2e 31 04 0 15 2e 32 55 2 25 35 2e 32 55 32 4e 54 3a 75 7; 99 66 69 61 66 76 66 66 69 6 67 6a 61 6c 66 66 66 69 6 67 6a 36 07 47 2e 32 2e 31 38 38 55 53 4 33 39 38 24 39 32	bytes cap 52:07:03:8: 1, Det: 2 1, Det: 2 1907: 1900 01:07 49:45:50 49:45:5	tured (2320 bits ), Dit: JPv4nca 9), Dit: JPv4nca 9), Dit: JPv4nca 9, 255.255.256 9 1, 1, 2, 1, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,	) st_7f:ff:fa (01: .E.  JFY Hos .25 n:s lia ce: .1. ve. .pi/ 00-	00:50:7f;ff;fa)		
<ul> <li>Frame</li> <li>Ethic</li> <li>Int:</li> <li>User</li> <li>Sim</li> <li>Octor</li> <li>Octo</li></ul>	me 1: 305 bytes ernet IT, Src: ernet Protocol r Datagram Proto Die Bervuce Din Die Bervuce Din Die Bervuce Din Die Bervuce Din Die	s on wire (25 Belkinn 07; Version 4, 1 50001, Src P, 500Very Prots 7 fa 94 44 1 0 60 84 11 0 60 84 11 9 20 32 35 0 30 00 88 1 7 6 0 14 9 45 59 27 9 20 32 35 0 30 00 88 1 7 4 65 69 27 1 7 3 20 77 1 7 3 20 77 1 7 3 20 77 1 7 3 20 4 1 7 4 5 66 6f 1 2 4 4 64 3 8 8 66 6c 94 8 9 30 38 38 6 39 30 38	$\begin{array}{c} 228 \ \text{birs}, \ 305 \\ \text{as:e} \ (94:44: \\ (94:44: \\ (75) \\ \text{col} \ 120, \ 306: \\ 100, \ 306: \ 306: \\ 100, \ 306: \ 306$	bytes cap 52:67:43:80 1, Dst: 2 Port: 100 00 45 00 01 ef ff 49 46 59 48 6f 73 2e 32 35 5e 3a 73 6e 30 51 3a 31 6d 76 65 60 61 3a 31 6d 76 63 65 3a 3a 31 6d 76 32 ef 38 30 2f 75 75 69 38 30 2f 35 32 2	tured (2329 bits a), Dst: TPv4mca 39, Dst: TPv4mca 29, 255, 256, 259 0 , 1, 1, K, .NO , 1, 1, K,, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	) t. 7f:ff:fa (01: .E.  Hos Lis Lis Lis Lis Ce: .1. .25 .25 .25 .25 .25 .25 .25 .2	88:56:7f:ff:fa)		
<ul> <li>France</li> <li>Ethy</li> <li>Inte</li> <li>Usee</li> <li>S1m</li> <li>00000</li> <li>0010</li>     &lt;</ul>	me 1: 305 byte: ernet I, Src: ernet I, Src: Bill Bervice Bill Bervice	s on wire (25 BelkinIn,07: Version 4, 2 toccol, Src Pr becwery Proto f fa 94 44 1 0 60 04 11 7 6c 01 4b 4 54 50 27 9 20 32 55 0 30 00 6a 1 7 3 24 77 1 7 24 77 3a c 41 44 31 1 7 46 9 6f 4 31 30 30 8 66 6c 04 1 0 30 30 30 4 31 30 30 8 93 30 8 95 8 93 30 8 95 8 95 8 95 8 95 8 95 8 95 8 95 8 95	228 bits), 305. 3316 (94:44) rrc: 192.188.2 rrt: 1940, 0st col 22 07 83 59 04 52 07 83 59 04 52 07 83 59 04 52 07 83 59 04 52 93 16 44 55 53 22 35 33 44 45 55 53 22 35 33 45 53 32 53 56 53 69 61 66 56 66 69 61 67 68 66 69 61 57 63 61 62 65 53 63 25 53 43 28 30 38 20 49 33 30 28 20 49 33 30 28 20 49 33 30 28 20 49 33 35 72 66 66 83 55 53 43 43 35 53 43 43 35 20 49 33 35 72 66 66 83 55 53 43 33 30 38 20 49 33 30 38 20 49 33 37 57 66 66 83 57 57 66 66 83 57 57 66 35 58 55 33 43 50 20 49 33 57 76 66 83 57 57 66 85 57 57 67 57 57 66 85 57 57 67 57 57 67 57 57 57 57	bytes cap 52:07:03:00 1, Det: 2; Port: 1900 00 45 60 61 ef ff 49 46 59 48 6f 73 20 32 35 60 49 46 59 48 6f 73 20 32 35 60 49 46 59 48 6f 73 36 20 61 63 65 30 33 31 04 76 55 66 93 30 32 41 36 30 24 36 36 25 54 55 54 55 55 54 55 55 55 55 55 55 55 55 55 55 55 55 55	tured (2329 bits 9), Dst: TPv4mca 39, 255, 255, 258 9 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -	) tt,7f:ff:fa (01: .E.  Hos Nos .25 n:s lia ce: .11. ve. .p;/ 00- 452 che 	B8:56:7f:ff:fa)		

Figure 4.3(b). Sniffed Results (Promiscuous Mode)

**Description:** The above figures show sniffing in managed/promiscuous mode. The sniffer is executed remotely via SSH and the results are sent back to the command center. The command center stores the result in a file which can be analyzed further.



# Monitor Mode Sniffing



Figure 4.3(c). Server and Client Execution

Appl	y a display filter									E	xpression	10.14
20			and the second second	and the second	A country in the							22.0
0.	Time .	Source	Destination	Prococol	Length inro	Contract of	THE OTHER	CN-0	Elaner.	87-100	CSTD: de	
	2.0.000000	Belkinin 07:43:09	Broadcast	802.11	185 Beacon	France	SN=255	EN=0	Elansz	BTut00	SSTDER	
	3 0.000000	BelkinIn 07:a3:89	Broadcast	802.11	185 Beacon	fraze.	SN=256.	FN=0.	Flags=	BI=100.	SSI0=dr	
	4 0,000000	BelkinIn 07:a3:89	Broadcast	802.11	185 Beacon	frame.	SN=257.	FN=0.	Flags=	BI=100.	SSID=dr.	
	5 0.000000	BelkinIn 07:a3:89	Broadcast	802.11	185 Beacon	frase.	SN#258.	FN=0.	Flegs#	BI=100.	SSID=dri	
	6 0.000000	BelkinIn 07:a3:89	Broadcast	802.11	185 Beacon	frame,	SN=259,	FN=0,	Flags=	BI=100,	SSID=dri	
	7 0.000000	BelkinIn_07:a3:89	Broadcast	802.11	185 Beacon	frame,	SN=260,	FN=0,	Flags=	BI=100,	SSID=dri	
	8 0.000000	BelkinIn_07:a3:89	Broadcast	802.11	185 Beacon	frane,	SN=261,	FN=0,	Flags=	BI=100,	SSID=dri	
	9.0,000000	BelkinIn 07:a3:89	Broadcast	802.11	185 Beacon	frame.	SN=262.	FN=0.	Flegs=	BI=100.	SSID=dri	
Radi 802. IEEE IEEE	totap Header v0 11 radio infor 802.11 Beacon 802.11 wirele 00 00 18 00 20	, Length 24 mation frame, Flags: ss LAN	), 185 bytes capt	ured (1400 bits)	v.							
Radi 802. IEEE IEEE 1000 0010 1020	0tap Header v0 11 radio infor 802.11 Beacon 802.11 wirele 00 00 18 00 22 a0 00 c5 00 00 ff ff 94 44 52	A Length 24 mation frame, Flags: ss LAN 48 00 a8 20 08 00 08 00 c5 00 80 00 08 107 a3 89 94 44 52	0, 185 bytes capt 00 00 02 76 09 00 ff ff ff ff 07 a3 89 e0 0f	@	v.							
Rad3 802. 1EEE 1EEE 1000 1010 1020 1030	0tap Header v0 11 radio infor 802.11 Beacon 802.11 wirele 00 00 18 00 2 a0 00 c5 00 00 ff ff 94 44 52 88 a2 a6 7f 2	Charle (140 010) mation frame, Flags: st LAN 40 00 a0 20 00 00 00 c5 00 80 00 00 00 c5 00 80 00 00 20 7 a1 89 94 44 52 10 00 00 00 66 40 11	00 00 02 76 09 00 ff ff ff ff 07 a3 89 e0 0f 04 00 6b 64 72		v.  dr							
Radi 802. IEEE IEEE 1000 1010 1020 1030 1040	otap Header v0 11 radio infor 802.11 Beacon 802.11 Wirele 00 00 18 00 22 a0 00 c5 00 00 ff ff 94 44 52 88 a2 a6 7f 22 61 67 6f 6e 62 90 40 c, 01 00	CL Length 24 mation frame, Flags: st LAN 140 00 a8 20 00 00 00 c5 00 80 00 00 07 a3 89 94 44 52 00 00 00 66 40 01 12 51 56 55 30 10 85 00 55 54 40 10 60	00 00 02 76 09 00 0f ff ff ff 07 a3 89 e0 0f 64 00 0b 64 72 82 84 8b 96 24		v.  dr							
Rad: 802. IEEE IEEE 1000 1010 1020 1030 1030 1050 1050	Lotap Header v0 11 radio infor 802.11 Beacon B02.11 Wirele 00 00 18 00 2 a0 00 c5 00 00 ff ff 94 44 55 88 a2 a6 7f 22 61 67 6f 66 63 30 48 6c 63 81 16 43 91 8 0	C Length 24 mation Frame, Flags: ss LAN 4 0 00 a0 20 00 00 00 c5 00 00 00 00 00 c5 00 00 00 00 00 c5 00 00 00 107 a3 89 94 44 55 00 00 06 64 00 11 51 56 56 5 80 1 08 103 65 94 06 11 06 06 09 01 6c 20 20	00 00 02 76 09 00 ff ff ff ff 00 a 80 00 76 09 00 ff a 80 00 04 00 06 64 72 02 84 80 96 24 00 2a 81 04 2f 06 00 0f ac 84	0. 0RDR agonball Z	v.  .s /							
Rad: 802. IEEE IEEE 1000 1010 1020 1030 1030 1050 1050 1050 1050 1050	Otap         Header         v0           11 radio         infor         802.11 Beacon           802.11 Wirele         00         00         18 urele           00         00         18 urele         00         00         18 urele           00         00         18 urele         00         00         18 urele           01         00         50         00         00         ff ff 94 44 52         18 a2 a6 7f 22           01         67         67         62         63         94 a6 cc 43 e1         91           02         67         64 a6 cc 43 e1         91         64 a6 28         90         07 ac 22         91	Length 24 msLion frame, Flags: ss LAN 40 00 a0 20 00 00 00 c5 00 80 00 00 00 07 3 38 9 44 45 22 00 00 00 64 64 00 11 20 10 c5 65 80 10 00 03 85 64 00 21 00 03 85 64 00 21 00 00 80 67 ac 02 20	00 00 02 76 09 00 ff ff ff ff 04 09 06 02 76 09 00 fg ff ff ff ff 04 09 06 64 72 02 84 05 96 24 06 7a 81 96 24 06 7a 64 26 22		v.  .s. 							
Radi 802. IEEE IEEE 1000 1010 1020 1030 1040 1050 1050 1050 1050 1050 1050 105	Lotap Header v0           11 radio infor           802.11 Beacon           802.11 Beacon           802.11 wirele           00 00 55 00 00           81 2 16 7 67 68           30 48 0c 03 11 61           81 04 30 18 01           81 04 30 18 01           81 04 30 18 01           81 04 30 18 01           81 04 30 18 01           81 04 30 18 01           81 04 09 00	Length 24 mallon frame, Flags: st LAN 48 00 a8 20 08 00 00 207 a3 80 94 44 52 00 00 66 00 80 00 00 207 a3 80 94 46 20 10 00 66 66 40 01 2 01 66 65 5a 01 08 03 85 94 60 21 00 08 80 67 ac 02 02 00 00 00 47 ac 02 02	00 00 02 76 09 00 ff ff ff ff 07 a3 89 60 07 22 84 6b 96 24 08 a8 96 04 08 28 4 6b 96 24 09 28 4 06 12 09 09 6f ac 84 09 38 46 12 09 09 6f al 80		v. 							
Radi 802. IEEE IEEE 1000 0010 1000 1000 1000 1000	Lotap Header v0 11 radio infor 802.11 Beacon 802.11 Wirele 00 00 15 00 2 a0 00 c5 00 00 ff ff 94 44 52 61 67 0f 66 3 30 48 6c 63 61 61 84 39 18 3 90 0f ac 02 00 18 60 dd 09 00 50 f2 81 e1 00	Length 24 malion frame, Flags: 5 LN 40 00 ml 20 00 00 07 ml 29 94 44 00 00 00 00 40 00 10 ml 20 00 00 06 00 00 44 00 11 61 66 5 ml 10 06 00 01 ml 20 20 00 00 01 ml 20 20 00 00 01 ml 20 20 10 18 92 02 02 00	00 00 02 76 09 00 ff ff ff ff 04 00 66 47 20 96 97 04 00 66 64 72 22 84 80 96 24 00 28 81 04 24 06 00 6f ac 64 06 32 04 0c 12 06 00 df ac 64 06 36 12 04 06		v. 							
Radi 802. IEEE IEEE 1000 1010 1020 1020 1020 1020 1020 1	Lotap Hender v0 11 radio infor 002.11 Beacon 802.11 virele 00 00 18 00 2 a0 00 05 00 00 17 ff 94 44 52 88 a2 a5 7f 22 61 67 6f 06 65 61 64 39 18 51 61 64 39 18 51 81 64 60 00 00 6f ac 02 61 18 60 d0 09 00 50 f2 21 81 00	Length 24           mation           frame, Flags:           st LAN           a0 00 all 20 00 00           00 c5 50 00 00 00           07 as 9 94 44 52           00 00 66 00 all 20 01 00           00 00 00 66 00 all 20 01 00           00 00 00 66 00 00           00 00 00 66 00 00           00 00 00 66 00 00           00 00 00 67 ac 02 00           00 00 00 67 ac 02 00           11 18 02 02 70 01           00 56 72 02 02 00           00 56 72 02 02 d0           00 56 72 02 02 d0	00 00 02 76 09 00 ff ff ff ff 07 a3 89 e0 07 40 09 66 47 22 52 84 8b 96 24 00 24 0c 12 00 08 ff c 14 27 00 00 e1 ac 04 00 32 e4 0c 12 00 08 df 18 10 08 56 f2 84 10		v. 							
Radi 802. IEEE IEEE 1000 1010 1020 1020 1020 1020 1020 1	Lotap Hender v0 11 radio infor 002.11 Meacon 002.11 Wirele 00 00 15 00 20 ff ff 04 44 52 86 a2 a6 7f 22 61 67 67 66 63 30 46 6c 83 62 16 44 30 18 61 60 86 ac 02 05 50 f2 62 01 00 4a 00 01 10 10	Length 24           mation           frame, Flags:           ad 00 ad 20 000           00 05 500           00 06 ad 20           00 06 ad 400           00 06 ad 400           00 06 ad 400           00 06 ad 400           00 06 ad 60           00 06 ad 60           00 06 ad 60           00 06 ad 61           00 06 ad 62           00 06           00 06	00 00 02 70 00 00 ff ff ff ff 04 00 60 02 70 00 04 00 60 61 73 28 48 60 90 22 04 00 40 64 72 28 48 60 90 24 00 28 01 64 25 00 00 61 62 04 00 28 40 12 00 09 60 41 20 00 90 60 61 20 00 60 61 61 61 00 60 61 00 60 61 00 61 61 00 61 00 61 61 00 61 00 00 61 00 00 61 00 00 61 00 00 61 00 00 00 61 00 00 00 61 00 000 000 000 0000000000		V. 							
Radi 802 1EEE 1EEE 000 010 820 000 820 000 000 000 000 000 000 00	Ottp         Hender         YO           11         radio         infor           602.11         Beacon         602.11         keacon           602.11         Wirels         602.11         keacon           600.61         802.11         wirels         602.11         keacon           600.61         802.11         wirels         602.11         keacon         602.11           800.61         81.62         802.11         keacon         602.11         602.11         602.11           88.22         86.71         72.62         61.67         62.62         61.64         60.61 <td>Length 24 Million Frame, Flags: Ss LAW 0 00 c5 00 80 00 00 00 0 00 c5 00 80 00 00 00 0 00 c5 00 80 44 652 0 00 00 60 64 00 11 0 10 c5 65 a0 108 0 00 60 67 ac 02 00 0 00 60 60 7 ac 02 00 0 00 65 67 20 20 20 00 0 05 56 72 02 04 00 44 00 01 02</td> <td>00 00 02 76 09 00 ff ff ff ff 07 a3 89 e0 ff 28 40 96 b6 4 72 82 44 80 96 24 80 24 81 44 2f 80 08 06 64 72 82 44 80 96 24 80 24 81 44 2f 80 09 61 62 24 40 90 32 64 46 12 90 99 64 21 44 60 90 59 f2 24 40 90 59 f2 44 10</td> <td></td> <td>v. dr </td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>	Length 24 Million Frame, Flags: Ss LAW 0 00 c5 00 80 00 00 00 0 00 c5 00 80 00 00 00 0 00 c5 00 80 44 652 0 00 00 60 64 00 11 0 10 c5 65 a0 108 0 00 60 67 ac 02 00 0 00 60 60 7 ac 02 00 0 00 65 67 20 20 20 00 0 05 56 72 02 04 00 44 00 01 02	00 00 02 76 09 00 ff ff ff ff 07 a3 89 e0 ff 28 40 96 b6 4 72 82 44 80 96 24 80 24 81 44 2f 80 08 06 64 72 82 44 80 96 24 80 24 81 44 2f 80 09 61 62 24 40 90 32 64 46 12 90 99 64 21 44 60 90 59 f2 24 40 90 59 f2 44 10		v. dr 							
Radi 802. IEEE 1000 000 000 000 000 000 000 000 000	Lotap Hender v0 11 radio infor 602.11 Beacon 802.11 wirele 00 00 18 00 22 00 00 05 00 00 ff ff 94 44 55 30 40 0c 03 00 f0 40 0c 03 00 60 40 0c 03 00 50 f2 20 10 00 50 f2 20 10 40 00 01 10 10	$\begin{array}{c} \mbox{Length} & 24 \\ \mbox{million} & \mbox{frame, Flags:} \\ \mbox{ssLAW} & \mbox{ssLAW} $	00 00 02 76 00 00 ff ff ff ff 07 33 39 60 ff 27 40 00 66 472 28 48 05 62 47 28 48 05 62 47 29 49 48 05 48 47 20 50 51 48 50 50 51 20 50 51 20 51 20 50 51 20 41 20 51 20 51 20 51 20 51		V. 							
Radi 802. IEE IEE 1000 0010 0020 0020 0050 0050 0050 0050	totap Hender v0 11 radio infor 802.11 Mencon 802.11 Wirele 00 00 15 00 20 ff ff 94 44 52 61 67 0f 06 63 90 0f c 63 92 61 64 30 18 00 00 60 0f c 62 01 18 60 dd 00 00 50 f2 21 01 00 4a 00 61 10 10	Length 24           million           frame, Flags:           ss LAN           48 06 m8         20 08 00           00 c5 00         80 00 00           00 c6 00         80 00 00           00 c8 00         80 00           00 00 0f         c0 20           10 18 02         02 100           00 00 0f         c0 20           10 18 02         02 00 00           00 00 0f         c0 20           14 00 50 72         60 d0	00 00 02 76 00 00 07 41 50 00 77 04 00 06 06 07 04 00 06 06 07 02 04 00 06 06 00 07 01 00 07 00 00 07 01 04 00 08 06 00 07 00 00 07 01 04 00 08 01 04 00 00 08 05 07 04 00 00 08 05 07 04 10	ured (1460 bits)	₩. 							

Figure 4.3(d). Sniffed Results(Monitor Mode)

**Description:** The above figures show sniffing in monitor mode. The sniffer is executed remotely via SSH and the results are sent back to the command center. The command center stores the result in a file which can be analyzed further.

## 4.4 Active attacks

Unlike passive attacks, active attacks involve some sort of interaction with the target system. There are many types of active attacks, ranging from port scanning to Remote Code Execution [4].

## **Port Scanning**

ubuntu@ubuntu:-/Pynthacker/Attacks/active/Scanning\$ python TCPScan.py cbit.ac.in	
TCP part 22 open	
TCP part 25 open	
TCP part 53 open	
TCP part 80 open	
TCP part 443 open	
TCP port 3386 open	
TCP port 41000 open	
TCP port 49787 open	
ubuntuğubuntu:-/Pynthacker/Attacks/active/Scaming\$	

Figure 4.4.(a) Available open ports

**Description:** Figure 4.4.(a) shows the results of a port scan of remote machine enumerating the open ports. These ports can be mapped to specific services and attacks can be carried further.

#### **Directed and Definite De-authentication**

	RX bytes:1058554619 (1009.5 MiB) TX bytes:12929710 (12.3 MiB)
wlanO \$	Link encap:UNSPEC HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-



N	lan.fc.type_subtype	0x9c					80	* Expression	1+	
	Time	Source	Destination	Protocol L	Length Info					
	4 9.386331696	Belkinin 97:a3:89	DnoplusT 81071:03	892.11	37 Deauthontication	\$14-3840	Field,	£1ags=		
	5 0.380392829	OmeplusT_01:71:03	BelkinIn_07:a3:09	892.11	37 Deauthentication	, SN=3849,	FN=0,	Flags=		
	5 8.389393433	BelkinIn_87:a3:89	OneplusT_01:71:03	892.11	37 Deauthentication	, SN=3849,	FN=8,	\$1ags=		
	7 0.380394138	OneplusT_01:71:03	BelkinIn_07:a3:89	882.11	37 Deauthentication	, SN-3840,	FN-0,	Flags		
	8 8.888384799	BelkinIn_07:83:89	OneplusT_01:71:03	892.11	37 Deauthentication	, SN=3849,	FN=0,	F1808=		
	9 0.080395247	OmeplusT_01:71:03	BelsinIn_97:a3:89	992.11	37 Deauthentication	, SN=3849,	FN=0,	Flags=		
	73 8.758435112	Belkinin_87:a3:89	Oneplus7_01:71:63	832.11	37 Deauthentication	, SN=3840,	FN=8,	Flags=		
	14 8.758457152	unepiusi 81:71:03	Selwinin 07:43:89	887.11	37 Deauthentication	, SN+3849,	FN-B,	+1ags		
	75 0.758488864	Belk1nIn_07:a3:89	Oneplus7_01:71:03	892.11	37 Deauthentication	, SN=3840,	FN=0,	F18Q9=		
	75 0.758484399	Onepius1_01:71:03	Beixinin_07:83:89	892.11	37 Deauthentication	, SN=3849,	FN=G,	F1898=		
	77 0.758407168	Belkinin_07:53:89	Onep1us1_01:/1:03	892.11	37 Deauthentication	, 5813840,	PAGE,	1 1 20152		
	7X 14 7582788847				THE RESIDENCE CONTRACTOR					
Fr Ramit	ane 4: 37 bytes o diotap Header v0, 2.11 radio inform EE 302.11 Deauth EE 302.11 wirelet	m wire (206 bits), Length 11 watton entication, Flags: . Is LAN	File Edit View ubuntuğubuntı y -a 94:44:54	ubuntu ubuntu Search Te :-/Pyntha 2:07:a3:89	37 Beauthentication gubuntu: ~/Pynthacke erminal Help acker/Attacks/acti 9 -c 94:65:2d:01:7	ve/Packe 1:03 wlx	/active, tinje 00e1b	rlans. /Packetinjeci ctions\$ su 01241fa	tions do pytho	an
FYRABIEI	ame 4: 37 bytes ( dictup Header v0, 27.11 radio inform 27.22 average ( 302.11 mirele) 202.11 mirele)	m wire (296 bits), Longth 11 arion wricerion, Flags: . s LAN	Religion 87-22 kg File Edit View ubuntugubunti y -a 94:44:57	ubuntu( Search Te at-/Pyntha 2:07:a3:89	37 Beauthentication gubuntu: -/Pynthacke erminal Help scker/Attacks/actt 9 -c 94:65:2d:01:7	ve/Packe 1:03 wlx	/active, tinje 00e1b	claina /Packetinjeci ctions\$ su 01241fa	tions do pytho	an
「大油印艺工」	ane 4: 37 bytes : diolap Header 90 7.31 ratio infor EE 882.11 Deauth EE 982.11 wirele 80 00 00 00 00 65 74 61 71 03	m wire (296 bits), Longth 11 arian writesricesrion, Flags: . s LAN 80 82 98 98 98 96 66 66 6 54 44 52 97 25 88 5	Baltinin 87-22 ka File Edit View ubuntuğubunt y -2 94:44:57	ubuntu( Search Te at-/Pyntha 2:07:a3:89	37 Beartbertration Gubuntu: -/Pynthacke erminal Help acker/Attacks/actt 9 -c 94:65:2d:01:7	ve/Packe	Active,	claina /Packetinjeci ctions\$ su 01241fa	do pytho	an
17 14 10 E E	ate 4: 37 bytes diotap Header 40 7:11 ratio anfor EE 882.11 Deanth EE 882.11 wirele 88 00 60 60 60 65 2d 61 72 63 89 00 70 67 00	m wire (286 bils), Length 11 witton etilcation, Flags: . s LAN 80 82 08 09 00 66 0 56 44 52 07 a3 85 5	Rolinin 87-224 File Edit View ubuntuğubunt y -3 94:44:5; 0 10 10 10 10 10 10 10 10 10	ubuntus search Te scarch T	37 Beartbertration Gubuntu: -/Pynthacke erminal Help acker/Attacks/actt 9 -c 94:65:2d:01:7	ve/Packe 1:03 wlx	tinje doelb	riade /Packetinjeci ctlons\$ su 01241fa	da pytha	on
FY AMORE	ate 4: 37 bytes dictap Header v0 21 11 ratio infor EE 882.11 Desurb EE 882.11 wirele 80 00 00 00 01 wirele 5 24 6: 17 03 80 00 T0 07 00	Wife (206 D12), Longh D 12(3), Longh D 12(3), Matter Matter Hilling C 206 00 00 00 S0 22 00 00 00 00 00 S0 44 52 07 25 80 5	Rolinin ar 22 ka File Edit View ubuntuğubuntı y -3 94:44:5;	ubuntu search Te s:-/Pyntha 2:07:a3:89	37 Bourthorfication gubunt:-/Pynthacke erminal Help ecker/Attacks/actt 9 -c 94:65:2d:01:7	ve/Packe 1:03 wlx	active,	Clair /PacketInject ctions\$ su 01241fa	da pytha	an
France Contraction	ane 4: 37 bytes dictap Header V0 2.11 ratio infor EE 882.11 Desurb EE 882.11 wirele 88 00 00 00 00 00 55 24 6: 07 03 88 00 70 07 03	m wire (206 Dits), Length 11 auton 11 metication, Flags: . s LAN 20 02 00 00 00 00 00 00 54 44 52 07 25 80 5	Ra Link ar 20 Se 7 by File Edit View ubantugbunt y -3 S41:44152 - 3 S41:44152 - 4 S4	822 11 ubuntu( : Search Te s:-/Pyntha 2:07:a3:89	27 Bourbert/sation glubunu: "Pynthacke erminal Help scher/attacks/actt 9 -c 94:65:2d:01:7	ve/Packe 103 wlx	/active, tInje 00e1b	Claire /Packetinject cttons\$ su 01241fa	do pytho	on
FF RaBIEL	ate 4: 37 bytes dictap Header 90, 21 1 ratio anter 22 802.11 Death 25 902.11 Death 26 90 90 90 90 90 65 2d 91 71 93 89 90 70 97 00	m wire (206 blts), Length 11 ation ation site (206 blts), respective site (200 blts), flags: . s LMS 80 b2 00 00 00 00 00 00 00 64 44 52 07 25 80 5	Ra (1764 27-20-26 77 by) File Edit: View obunt utglubant y - a \$41:64:52	822 11 ubuntu( Search Te 12-/Pyntha 2:07:a3:89	27 Bourhart (attoo gubunu: "Pynthack eminal Help acker/Attacks/actt	ve/Packe 1:03 wix	/active, tinje 00e1b	/Packetinjeci /Packetinjeci ctions5 su 01241fa	do pytho	on i

Figure 4.4(c). De-authentication Packets

	collisions:0 txqueuelen:1000
air	RX bytes:10585/6995 (1009.5 M1B) IX bytes:12943282 (12.3 M1B)
wlan0	Link encap:UNSPEC HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-
	UP BROADCAST WULTICAST MTU:1500 Metric:1
	RX packets:1425568 errors:0 dropped:0 overruns:0 frame:0
	TX packets:678232 errors:0 dropped:0 overruns:0 carrier:0
	collisions:0 txqueuelen:3000
	RX bytes:1848067489 (1.7 GiB) TX bytes:112967838 (107.7 MiB)
s 📕	

## Figure 4.4.(d). Targeted Device after De-authentication

**Description:** The above figure shows the result of performing a targeted deauthentication attack on a station with definite number of deauthentication packets.

# VI. CONCLUSION AND FUTURE SCOPE

A drone has been built which is controlled by and carries with a Linux machine. The drone is augmented with the capability to interact with the network environment around it, so it can survey, recon or attack the network infrastructure around it. A library of required structures and



procedures to attack the networks has been developed and deployed, which can be customized according to need. A few attacks have been implemented which demonstrate the proof of concept of customizing attacks. This paper has investigated the possibility of unifying disjoint domains of automation and robotics, with network security.

## **FUTURE SCOPE**

This work explores the idea of providing locomotion to computers by augmenting them with motion peripherals. Besides, this also has the capability of network surveillance and attacks. This is just one variation of what can be done and how functionality of computers can be extended by providing locomotion to them. The amount of scope for future works is only limited by creativity of individuals. Many improvements can be made such as creating dedicated chip sets (application specific integrated circuits) which can be used to speed up the execution of required objective. Instead of an aerial reconnaissance vehicle, attempts may be made to develop their equivalents over land and also under water. From the software perspective, multithreaded programming can be implemented to leverage the full power of today's multicore processors.

# REFERENCES

- [1] A Survey on Wireless Security: Technical Challenges, Recent Advances, and Future Trends, Yulong Zou, Jia Zhu, Xianbin Wang, Lajos Hanzo, Published in: Proceedings of the IEEE (Volume: 104, Issue: 9, Sept. 2016)
- [2] A SURVEY OF WIRELESS NETWORK SECURITY, S. Gopalakrishnan, Published in The International Journal of Computer Science and Mobile Computing, Vol. 3, Issue. 1, January 2014
- [3] Modelling and control of quadcopter, Teppo Luukkonen, Aalto University, School of Science, Independent research project in applied mathematics.
- [4] M. Whitman and H. Mattord, Principles of Information Security, 4th ed. Independence, KY, USA: Delmar Cengage Learning, 2012.
- [5] Y. Xiao, H.-H. Chen, B. Sun, R. Wang, and S. Sethi, "MAC security and security overhead analysis in the IEEE 802.15.4 wireless sensor networks," EURASIP J. Wireless Commun. Netw., 2006, doi: 10.1155/WCN/2006/93830.
- [6] G. Apostolopoulos, V. Peris, P. Pradhan, and D. Saha, "Securing electronic commerce: Reducing the SSL overhead," IEEE Network, vol. 14, no. 4, pp. 8–16, Jul. 2000.
- [7] K. Wong, Y. Zheng, J. Cao, and S. Wang, "A dynamic user authentication scheme for wireless sensor networks," in Proc. IEEE Int. Conf. Sensor Netw. Ubiquitous Trustworthy Comput., Taichung, Taiwan,

Jun. 2006, doi: 10.1109/ SUTC.2006.1636182, pp. 244–251.

- [8] A. Aziz and W. Diffie, "Privacy and authentication for wireless local area networks," IEEE Pers. Commun., vol. 1, no. 1, pp. 25–31, Aug. 2002.
- [9] G. Raju and R. Akbani, "Authentication in wireless networks," in Proc. 40th Annu. Hawaii Int. Conf. Syst. Sci., Waikoloa, HI, USA, Jan. 2007, doi: 10.1109/HICSS.2007.93.
- [10] L. Venkatraman and D. P. Agrawal, "A novel authentication scheme for ad hoc networks," in Proc. IEEE Wireless Commun. Netw. Conf., Chicago, IL, USA, Sep. 2000, pp. 1268–1273.
- [11] A. H. Lashkari, K. Lumpur, M. Mansoor, and A. S. Danesh, "Wired equivalent privacy (WEP) versus Wi-Fi protected access (WPA)," in Proc. Int. Conf. Signal Process. Syst., Singapore, May 2009.
- [12] K. J. Hole, E. Dyrnes, and P. Thorsheim, "Securing Wi-Fi networks," Computer, vol. 38, no. 7, pp. 28–34, Jul. 2005.
- [13] RFC 5246, "The Transport Layer Security (TLS) Protocol Version 1.2", Aug. 2008.[Online].Available: https://tools.ietf.org/ html/rfc5246
- [14] RFC 4346, "The Transport Layer Security (TLS) Protocol Version 1.1", Apr. 2006.[Online]. Available: https://tools.ietf.org/ html/rfc4346
- [15]RFC 2246, "The Transport Layer Security (TLS) Protocol Version 1.0", Jan. 1999. [Online]. Available: https://tools.ietf.org/ html/rfc2246
- [16] Mitchell Ashley , "A Guide to Wireless Network Security" Information systems Control Journal ,Volume 3,2004.
- [17] Karen Scarfone, Derric Dicoi, "Wireless Network SecurityforIEEE 802.11a/b/g,Bluetooth(DRAFT)",NISTPublication-800-48.Augest 2007.
- [18] Tom karygiannis, Les Owens, "Wireless Network Security for IEEE 802.11a/b/g,Bluetooth(DRAFT)",NISTPublication-800-48.November 2002.
- [19] Ahmed M. Al Naamany , Ali Al Shidhani, Hadj Bourdoucen, "IEEE 802.11 Wireless LAN Security Overview", IJCSNS International Journal of Computer Science and Network Security, VOL.6 No.5B, May 2006.
- [20] "Cryptography and Network Security" By William Stallings
- [21] http://www.instructables.com/id/The-Pi-Quadcopter