International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 7– No.11, November 2014 – www.ijais.org

Cracking and Hardening Hidden SSID Mechanism in 802.11 using PYTHON

Varun Chopra
Department of Computer Science and Engineering,
Maharaja Agrasen Institute of Technology,
Delhi

Sushil Mehra
Department of Computer Science and Engineering,
Maharaja Agrasen Institute of Technology,
Delhi

ABSTRACT
Wireless Protocol has become a very important part of modern life. From cell phone to laptops everyone uses Wireless for one purpose or another. Wireless Protocol is commonly known as IEEE 802.11 Cliendard. It Uses Several Frames, Header for its operation. Unfortunately none of these are encrypted and hence can be sniffed and analyzed for exploiting. Due to this plaintext nature of 802.11 packets several protection mechanisms provided by 802.11 become useless such as hidden ssid Mac filtering etc. In this paper, such protection mechanisms are analyzed thoroughly and it is explained how python can be used to defeat this protection schema, it is also shown how python can also be used for hardening these security measures.

General Terms
Python, Scapy, Security, Attacker, Packet

Keywords
Hidden, Wifi, SSID, Scapy, 802.11

1. INTRODUCTION
802.11[1] is IEEE Standard for the wireless lan used today by nearly every person on the earth. 802.11 Standard allows users to access internet, transfer files without any physical links through radio waves. 802.11 receivers/Transmitters are incorporated in every device we use that is cell phones, laptops etc. Transmission speed of 802.11 can reach up to 300 mbps 802.11 is generally known to common masses as "Wi-Fi" .To accomplish its tasks 802.11 uses several management and operational frames ,the structure and need for these frames will be covered in later part of paper. Due to its widespread adoption several protection schemes were implemented in 802.11 such as Wep which suffered from a logical flaw and hence could be cracked in minutes,WPA2 but these schemes really relied on strong passphrase for key generation and authentication ,otherwise the passphrase can be brute forced to compromise the network. To augment WPA2 mechanism , some other schemes such as hidden ssid,mac address filtering were introduced. This Paper focuses on logical flaws of 802.11 which render these additional protection mechanisms useless.

2. ORGANIZATION OF THE PAPER
This paper is organized as follows. Section 2 explains the working of 802.11 and various frames and messages used during operation. Section 3 deals with study of various 802.11 protection mechanisms used to provide security to users. Section 4 analyzes one of the security measures and explains how it provides a false sense of security to users. Section 5 explains how to defeat this protection mechanism using Python. Section 6 proposes a technique to harden the security measures. Section 7 enlists the future scope and conclusion. Section 8 acknowledges the mentor for his conClientnt support and section 9 lists the references used.

3. 802.11 WORKING AND VARIOUS PACKETS INVOLVED
A typical 802.11(wireless) setup consists of one or more access points and several clients. An Access Point is a device that allows several wireless clients to connect to wire internet connection. Mostly Access point is a wireless Router .It minimizes the need of cables in a network .For example to connect say 5 clients to internet,5 cables would be needed but by using Wireless setup only one wire is needed which would go into the router and all wireless clients can access this internet using the wireless connection from the router

.Authentication is needed in 802.11 to save the network from unauthorized usage and protect users file hosted on network.

Authentication Procedure is illustrated in figure 1.As inferred from the figure there are several messages exchanged during the authentication process; these messages play a very important role in proper functioning of the system. These messages are exchanged in form of packets. If all the messages are exchanged in the order as shown in the figure then client is successfully authenticated and is able to use to Access Point Services.

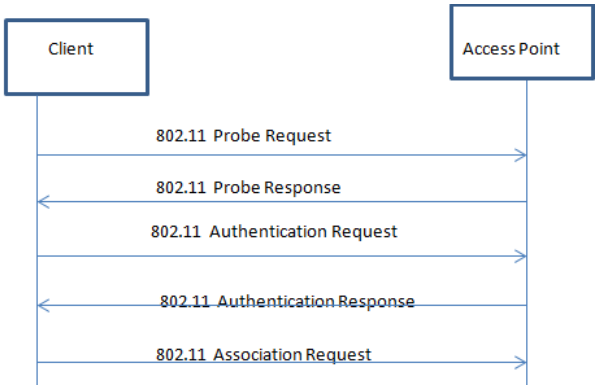

Figure 1. Shows messages exchanged during the authentication and association Process

Various Messages (Packets) transmitted [3] during the process are as follows:

A.)Probe Request Packet: The wireless client or CLIENT (Clienttion) searches the network to check if any APs are available. This is done by sending probe request packets. This packet is sent from CLIENT to AP. The client loops through various channels (hopping) and sends the probe request. The client can send a broadcast probe request or specific probe

request. In specific probe request either the user enters the SSID or the client uses the previous history or cache of the SSIDs it has connected to in past.

If the probe request packet is a broadcast packet then it is known as Null Probe Request Packet. This packet is sent by CLIENT to broadcast its presence to the AP and to ask them for their SSIDs.

B.)Probe Response Packer: An AP responds to the probe request of the client by probe response and tells the client that it is an valid AP and tells client it`s SSID.This packet also include access point's security parameter that is what kind of security access point uses and so on.

C.)Authentication Request Packet: In this packet an authentication request packet is sent from the client to AP, to authenticate or verify the connection. In this packet client sends the security key or passphrase which is used for authentication. An authentication response packet is sent from the AP to client telling the client if the authentication is successful or not,

The authentication request and response packets are always in sequence.

For example.

If the authentication request packet SEQ = 0x0001

Then the authentication response packet SEQ = 0x0002

D.)Association Packets: After the authentication is successful by the AP, then CLIENT sends an association request packet. This packet requests the AP to associate or tell the AP to make a connection to share data. The AP responds back to this request with association response packet and AP and CLIENT are associated now i.e. they are now connected and ready to share data.

Association response packets contain the parameters and capabilities of the connection.

E.)DEAuthentication Packet:This packet is used to deliberately break a connection. This packet can be sent by STA or AP. When the user wants to disconnect the wireless connection and clicks on disconnect option then STA sends a DeAuth packet which breaks the connection. Suppose there is some changes that are made on AP regarding the security or SSID then AP sends DeAuth packet to break all the connections to various STA. There is no intelligence in access point which can distinguish whether is deauthentication packet is by genuine client or by an attacker so this allows an attacker to forge deauthentication packet and inject it in mainstream using a tool and cause Denial of service attack.

3.1 802.11 FRAME FORMAT AND BEACON FRAMES

802.11 use the frame format shown in figure 2.

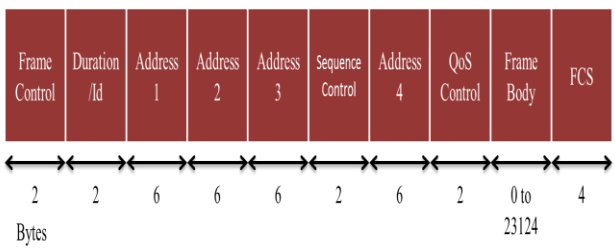

Figure 2. Frame Format used by 802.11

As Shown in figure a frame[3]-[3] format contains several fields which are used to control various operations of 802.11.Explanation Of these fields are as follows:

A.)Frame Control: Frame control field is of 2 bytes. It encapsulates various information such as protocol, type of packet (management, data or control), subtype of packet (authentication request, response etc.).

B.)Duration/ID: This field is also of 2 bytes. It can be used as duration or ID field in different circumstances. It is used to set NAV (Network Allocation Vector).

C.)Address fields: There are 4 address fields defined in the frame. It is not compulsory to use all 4 of them at one time. For instance in Wireless Lan ,3 address fields are used whereas in WDS 4 address fields are used. These Fields are of 6 byte each.

D.)Sequence Control: This field is used for controlling sequence of packets so packets arriving at destination can be identified

E.)Frame Body: Frame body contains the actual payload. It can contain the actual data that you are sending. It can also have the management header of the subtype of the packer.

F.)FCS: It stands for Frame Sequence Check. It is basically CRC (cyclic redundancy check) over the entire MAC header and frame body. What happens is, CRC is run over the entire MAC header with frame body and the four byte value is stored in FCS.

3.1.1 Beacon Frames

1.) Beacon Frames: Beacon frames are one of the manage-ment frames used by 802.11 for proper operation. These frames are broadcasted periodically by Access point to an-nounce its presence to all the wireless clients in its vicinity.

Due to this broadcasting of frames, we are able to see the wire-less SSIDS when we try to connect to a network. Beacon frame contains various fields for operation, some of them are:

a.)Time Stamp:After receiving beacon frame all stations change their local clock to this time,this is used for synchro-nization purposes.

b.)Beacon Interval:This is the time interval after which bea-con frame is broadcasted again.This allows station to go in power saving mode and wakes up again when there is time for receiving again.

c.)SSID-This is the name by which access point is identified by the station. This SSID can be changed by Modifying value in access point. Hidden SSIDS have value 0 in this feild

The main logic flaw that was observed while analyzing the packets of 802.11 was that these packets are transmitted in plaintext, so any malicious user can sniff and read the frames to uncover a vulnerability or for Ddos attack as there is no intelligence to find out whether any packet is authentic or a forged one by attacker. Due to this flaw any attacker can forge a deauthentication packet and disconnect all legitimate clients from access point.

4. 802.11 PROTECTION MECHANISMS:

As, Discussed earlier 802.11 incorporates several protection mechanisms for providing security[4] to the users.Some of these protection measures are discussed below:

A.)Shared Key Based Authenctication : In this protection scheme,a user needs to enter a correct key in order to get authenticated and use network services.Key can be numeric or text based or combination of both.Security of this protection mechanism depends upon type of encryption used by Access Point and strong passphrase key.Several encryption methods were used to protect the key from being derived from packets

B.)MAC ADDRESS FILTERING:MAC Address filtering is a protection feature in routers which allows network administrator to set certain rules for accessing the network based on mac address of device. This is an easy to break protection schema as an attacker can easily find out the list of mac address associated with access point by analyzing the packets and spoofing the mac address of malicious machine.

C.)Hidden SSID:Hidden SSID is a feature by which ssid of the access point is not broadcasted to users,instead associating client should know the ssid of the network explicitly and should enter that ssid while connecting to the network along with security parameters.This feature is accomplished by setting the ssid field to zero in the beacon frames.

4.1 HIDDEN SSID

As discussed earlier this feature allows users to hide the ssid of their networks so that there can be no hacking attempt on the network. This is done by setting the ssid field in beacon frame to zero. But this feature is very easy to bypass for an attacker because even if ssid is hidden then the ssid is transmitted in plain text in probe request and probe response packet, that means attacker can analyze probe request packets through manual analysis or through automated tools such as aircrack suite of tools to get the ssid and hence defeating this protection schema.So even if user is using hidden ssid feature for that network ,the network can be hacked or can be subjected to denial of service attack.

Figure 3 and 4 shows how hidden ssid is cracked within minutes:

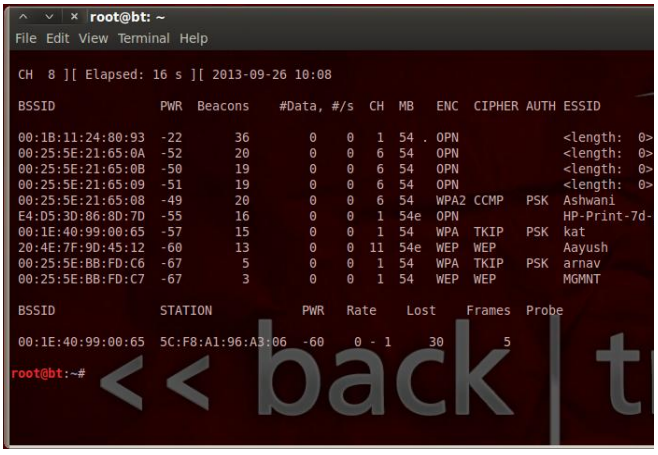

Figure 3. showing hidden ssids in the vicinity, length 0 fields corresponds to network with hidden ssids.

Now as soon as some client connects to access point ,the airmon suite tool will analyze the probe response packet and will display the hidden ssid:

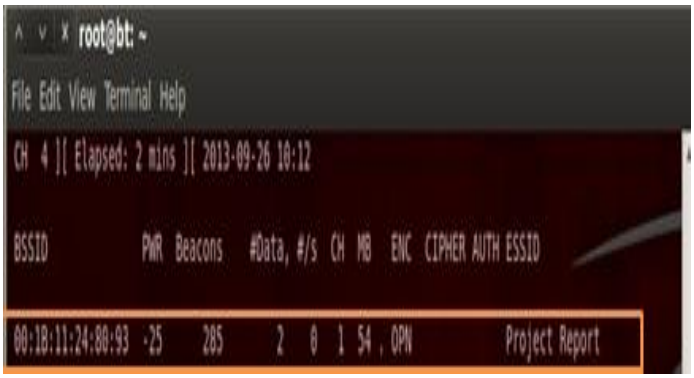

Figure 4. Showing uncovered hidden ssid

5. DEVELOPING HIDDEN SSID DETECTION TOOL IN PYTHON

Python[6] is a high level language which is easy to learn and has easier syntax than other high level or low level languages. Python supports multi-threading, object oriented programming, automatic memory management and has large number of libraries to accomplish various types of task. Hence Python is most preffered language to automate pentesting of 802.11 networks.

One Such module that exist in python is scapy,which will be used throughout the text to demonstrate application of python to 802.11.

Scapy [7]is packet manipulation program in python,which can forge or decode packets,sniff the packets ,send the packets on air/wire,match request and replies.It can also handle task such as network scanning,tracerouting,attacking[7] network etc.

By using scapy a pentester can write interactive scripts in such a way that they can incorporate multiple tasks into one script hence saving time and promoting efficiency.

Hence,Rather than using airmon suite of tools ,which include re-membering complex syntax and needs several input parameters such as mac address of access point ,channel of access point etc,It is feasible to harness the power of python and in few lines of code write a script to uncover hidden ssids.

5.1 Layering in Scapy

In order to understand the logic behind the program , a person must understand how a packet is arranged in different layers in scapy.. Any 802.11 packet captured through scapy can be viewed. A packet has several layers into it and each layer can have sub-layers.Each layer contains some information about packet and network.This information is analysed while writing scripts.

Linux operating system will be used for all code purposes.

International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 7– No.11, November 2014 – www.ijais.org

Figure 5 displays an 802.11 packet captured through scapy

```
>>> pkt
<RadioTap  version=0 pad=0 len=26 present=TSFT+Flags+Rate+Channel+dBm_AntSignal+Antenna+b14 notdecoded='\x01\x9a\x90b\x00\x00\
x00\x00\x10\x02l\t\xc0\x00\xbe\x01\x00\x00' |<Dot11  subtype=0L type=Management proto=0L FCfield= ID=0 addr1=ff:ff:ff:ff:ff:ff
 addr2=90:72:40:18:50:ec addr3=90:72:40:18:50:ec SC=25216 addr4=None |<Dot11Beacon  timestamp=949424128389L beacon_interval=10
0 cap=res8+short-slot+res12+ESS+privacy |<Dot11Elt  ID=SSID len=9 info='Alex WiFi' |<Dot11Elt  ID=Rates len=8 info='\x82\x84\x
8b\x96$0Hl' |<Dot11Elt  ID=DSset len=1 info='\x01' |<Dot11Elt  ID=TIM len=4 info='\x02\x03\x00\x00' |<Dot11Elt  ID=7 len=6 inf
o='TW \x01\x0b\x14' |<Dot11Elt  ID=32 len=1 info='\x00' |<Dot11Elt  ID=35 len=2 info='\x10\x00' |<Dot11Elt  ID=ERPinfo len=1 i
nfo='\x04' |<Dot11Elt  ID=RSNinfo len=20 info='\x01\x00\x00\x0f\xac\x04\x01\x00\x00\x0f\xac\x04\x01\x00\x00\x0f\xac\x02\x00\x0
0' |<Dot11Elt  ID=ESRates len=4 info='\x0c\x12\x18`' |<Dot11Elt  ID=45 len=26 info='\xad\x19\x17\xff\xff\xff\x00\x00\x00\x00\x
00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00' |<Dot11Elt  ID=61 len=22 info='\x01\x08\x15\x00\x00\x00\x00\x0
0\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00' |<Dot11Elt  ID=127 len=8 info='\x00\x00\x00\x00\x00\x00\x00@' |<Dot
11Elt  ID=vendor len=11 info='\x00\x17\xf2\x01\x00\x01\x01\x00\x00\x00\x07' |<Dot11Elt  ID=vendor len=7 info='\x00\x03\x93\x01
w\x03 ' |<Dot11Elt  ID=vendor len=14 info='\x00\x17\xf2\x07\x00\x01\x01\x06\x90r@\x18P\xed' |<Dot11Elt  ID=vendor len=9 info='
\x00\x10\x18\x02\x03\x00\x1c\x00\x00' |<Dot11Elt  ID=vendor len=24 info="\x00P\xf2\x02\x01\x01\x80\x00\x03\xa4\x00\x00'\xa4\x0
0\x00BC^\x00b2/\x00" |<Dot11Elt  ID=70 len=5 info='\x02\x00\x01\x00\x00' |<Dot11Elt  ID=178 len=63 info='\xa9\xcc' |>>>>>>>>>>
>>>>>>>>>>>>>
>>>
```

Figure 5. Showing a packet caputed through scapy

The red highlited text shows the layers and remaining part shows the information contained in that layer.

For example the above caputed packet is a beacon frame as evident by <DOT11BEACON> Layer. SSid is contained in <dot11elt> layer .In this case ssid field is set to zero hence the network must be using hidden ssid feature
Addr Field Gives the Mac Address Of the access point from which packets are originating.
Use ls(layer name) to gather more information about a layer

Code to Uncover all Hidden ssids:

```
#!/usr/bin/python                    /python shell on linux

From scapy.all import *          /import scapy libraby

Import socket                    /import socket library

Hidden=set()        /creating a set named hidden

Def Packethandler(pkt):
    If pkt.haslayer(Dot11Beacon) :
        If not pkt.info:
            If pkt.addr3 not in Hidden:

                hidden.add(pkt.addr3)
                Print "Hidden SSID NETWORK FOUND ! BSSID :",pkt.addr3

    Elif pkt.haslayer(Dot11ProbResp) and (pkt.addr3in hidden)
                Print "Hidden SSID UNCOVERED !",pkt.info,pkt.addr3

Sniff(iface="mon0",count=10000,prn=PacketHandler)
```

The above code is a customized tool to detect and list hidden ssids by examining the probe response packet.The code works as follows:

1.)A set is initialized named hidden in which all hidden ssids will be stored. This is done in order to stop listing the same hidden ssid again

2.) A Packet handler is defined in which packets sniffed will be parsed and analyzed.

3.)In Packet handler,first it is checked whether the packet is beacon frame or not.If it is beacon frame then it is checked whether it's ssid length is zero and if address of the ap is already available in set or not.If it is not available then that mac address of ap is added to list.

4.)If packet is not an beacon frame,then it is checked whether packet is proberesponse packet .If it is a probe response packet then info field in <dot11elt> is analyzed and that network is listed provided the mac address of that ap is in hidden set

5.)sniff function is used to start capture of the packet.iface is used to tell which interface to start sniffing on.For capture of packets ,wireless card should be put into monitor mode using airmon-ng start "interface name" command. Replace interface name by the interface of wireless card. Determine interface using iwconfig command. Count is used to specify no of packets to be captured and prn is used to parse the packets captured to packet handler.

Figure 6 shows the output of the code :

```
HIDDEN SSID Network Found! BSSID:  00:24:82:65:14:59
HIDDEN SSID Network Found! BSSID:  00:22:7f:29:71:09
HIDDEN SSID Network Found! BSSID:  00:22:7f:69:71:09
HIDDEN SSID Network Found! BSSID:  00:24:82:25:14:59
HIDDEN SSID Network Found! BSSID:  ac:67:06:34:7f:b9
HIDDEN SSID Network Found! BSSID:  00:22:7f:a9:71:08


HIDDEN SSID Uncovered!  this-is-secret 84:c9:b2:6c:63:03
HIDDEN SSID Uncovered!  this-is-secret 84:c9:b2:6c:63:03
HIDDEN SSID Uncovered!  this-is-secret 84:c9:b2:6c:63:03
HIDDEN SSID Uncovered!  this-is-secret 84:c9:b2:6c:63:03
HIDDEN SSID Uncovered!  this-is-secret 84:c9:b2:6c:63:03
HIDDEN SSID Uncovered!  this-is-secret 84:c9:b2:6c:63:03
HIDDEN SSID Uncovered!  this-is-secret 84:c9:b2:6c:63:03
HIDDEN SSID Uncovered!  this-is-secret 84:c9:b2:6c:63:03
HIDDEN SSID Uncovered!  this-is-secret 84:c9:b2:6c:63:03
HIDDEN SSID Uncovered!  this-is-secret 84:c9:b2:6c:63:03
HIDDEN SSID Uncovered!  this-is-secret 84:c9:b2:6c:63:03
HIDDEN SSID Uncovered!  this-is-secret 84:c9:b2:6c:63:03
HIDDEN SSID Uncovered!  this-is-secret 84:c9:b2:6c:63:03
```

Figure 6. Showing output of python code

6. PROPOSED METHOD FOR HARDENING HIDDEN SSID DETECTION

In this Section ,a technique is proposed to harden the hidden ssid protection that is making detection of hidden ssid difficult.

Most of the automated tools rely on probe responses to uncover hidden ssid of a network. As 802.11 does not have any mechanism to check whether a packet in the stream is forged or authentic. This lack of checking can be used to harden the hidden ssid detection by spoofing the probe response packets of the networks. This will be enough to thwart the attempts of script kiddies to penetrate the network.

Now we built a hidden ssid clocker which can cloack hidden ssid further. This cloacker can be built in few lines of code and can be made to run for any length of time in order to cloack the hidden ssid

The principal behind the code is that any forged [8]-[9] packet can be injected into wireless stream without any checking so in this case python script takes advantage of this flaw and forges a fake probe response packer and injects in the mainstream. This forged packet fools the automated tools into thinking that hidden ssid has been discovered but in reality only the spurious ssid value which was inserted in forged packet is discovered not the real one.

Code:

```
from scapy.all import *

mymac=”aa:bb:cc:dd:ee:ff”

apmac=”ab:cd:ef:ff:aa:bb”

pkt=RadioTap()/Dot11(type=0,subtype=5,addr1=mymac,addr2
=apmac,addr3=apmac)/Dot11ProbeResp()/Dot11Elt(ID=0,info
=”Cloacked”)/Dot11Elt(ID=1,info=”\x02\x04\x0b\x16”)/
sendp(pkt,iface=”mon0”,count=3,inter=.3)
```

The above code works by defining two variables in which one client’s and access point’s mac address is stored. After that a packet is forged .Packet is Probe response packet as defined in the packet and subtype=5 also defines a packet as probe response.Dot11Elt contains the information about the spurious ssid which is feeded to automated tools ,in this case spurious ssid is cloacked.Rest of the fields are for network rate,information.These can be filled by first capturing a network packet and analyzing it as it was shown in layering in scapy module

7. CONCLUSION

802.11 ieee protocol was thoroughly analyzed in a most basic way and its basic way of working was explained. It’s security mechanisms were discussed along with the reason of their shortcomings. It was also discussed how to script you own tools to crack as well as harden a particular 802.11 protection scheme known as hidden ssid.In future there is vast scope in this field to develop state of art customized tools for pen testing and securing 802.11 networks.

Difference between automated tools and python scripted tools.

- No of parameters specified in automated tools are far more than no of parameters in python scripted tools. For example base station mac addresses have to be specified in automated tools.
- Automated tools have difficult syntax to work with. Lots of options with “-“which can be troublesome for a starter.
- Automated tools can be spoofed whereas customization can be done in python scripts to make them more efficient

Advantages of proposed Method:

- Spoof the automated tools which give a false impression to attacker that he has cracked the network.
- Attacker has to resort to the manual methods of attacking the hidden ssid which can take longer and is more tedious job to do
- Efficient system in protecting the networks from so called script kiddies.

8. ACKNOWLEDGEMENT

We owe a special debt of gratitude to Asst. Professor Pooja Gupta, Department of Computer Science & Engineering, Maharaja Agrasen Institute of Technology, Delhi for her constant support and guidance throughout the course of our work. We would also like to thank Mr Vivek Ramachandran (Founder and CTO Of Penteser Academy) for teaching us the basics of 802.11 through his videos. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us.

9. REFERENCES

[1] Introduction to 802.11 Technology – Cisco (http://www.cisco.com/web/TH/assets/docs/seminar/200903_Night_Academy_3_1_WLAN_Introduction.pdf)

[2] IEEE 802.11 Wireless LANs (http://inst.eecs.berkeley.edu/~ee122/sp07/80211.pdf)

[3] IEEE 802.11 frame format - Studio Reti (http://www.studioreti.it/slide/802-11-Frame_E_C.pdf)

[4] Information Embedding in IEEE 802.11 Beacon Frame (http://research.ijcaonline.org/ctngc/number3/ctngc1027.pdf)

[5] An Overview of 802.11 Wireless Network Security Standards (http://www.sans.org/reading-room/whitepapers/wireless/overview-80211-wireless-network-security-standards-mechanisms-1530)

[6] Hands on python-Begginers (http://www.nervenet.org/pdf/python3handson.pdf)

[7] Scapy documentation (!) - SecDev.org (www.secdev.org/projects/scapy/files/scapydoc.pdf)

[8] Packet Crafting Using Scapy-William Zereneh (http://www.scs.ryerson.ca/~zereneh/cn8001/CN8001-PacketCraftingUsingScapy-WilliamZereneh.pdf)

[9] Network packet forgery with Scapy – PacSec (https://pacsec.jp/psj05/psj05-biondi-en.pdf)