KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY, KUMASI

-

-

VULNERABILITY ANALYSIS IN WIRELESS LOCAL AREA NETWORKS: A SURVEY OF SOME WIRELESS ACCESS POINTS IN GHANA.

-

By

-

KWABENA AKOMEA-AGYIN (MSc. Information Technology)

PG6564011

-CENTRE: ACCRA

2016

VULNERABILITY ANALYSIS IN WIRELESS LOCAL AREA NETWORKS: A SURVEY OF SOME WIRELESS ACCESS POINTS IN GHANA

-

BY

-

-

KWABENA AKOMEA-AGYIN (MSc. Information Technology)

PG6564011

A THESIS SUBMITTED TO THE INSTITUTE OF DISTANCE LEARNING, KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY IN PARTIAL FUFILMENT OF THE REQUIREMENT FOR AN AWARD OF MASTERS DEGREE IN INFORMATION TECHNOLOGY.

November, 2016

DECLARATION

I hereby declare that the submission of this compilation is the true findings of my own researched work presented towards an award of a second degree in Masters in Information Technology and that, to the best of my knowledge, it contains no material previously published by another person nor submitted to any other University or institution for the award of degree except where due acknowledgement has been made in text. However, references from the work of others have been clearly stated.

Kwabena Akomea-Agyin (PG6564011)	………………….	………………………
Student Name & ID	Signature	Date

Certified by:

Dr. Michael Asante	………………….	………………………
Supervisor's Name	Signature	Date

Certified by:

Dr. J.B Hayford-Acquah	………………..	…………………
Head of Dept. of Computer Science	Signature	Date

DEDICATION

I dedicate this work to my father Capt.(RTD) Samuel Kofi Akomea-Agyin and my mother Mrs. Theresa Akomea-Agyin for their love and support throughout my academic pursuit.

I also dedicate this work to every student and researcher who is undertaken researches within the fields of Network Security and Cryptography.

ACKNOWLEDGEMENT

I am grateful to my creator (Jehovah God Almighty) for His knowledge, grace, and strength that sustained me throughout the writing of this thesis work, thereby making it a success.

My second thanks go to my project supervisor, Dr. Michael Asante, who despite his busy schedules always made time for me, guided me on my research path, notified and always ensured I corrected all my research errors. I have done a good research that I will always be proud of for the rest of my life, all thanks to him.

My third appreciation goes to my brothers, sisters, friends, and co-workers around the globe who in one way or the other made my thesis work a success.

ABSTRACT

Wireless Communications can be found everywhere including banks, telecommunication companies, hotels, hospitals, academic institutions, government sectors, intelligence organizations, and the military. If these wireless communications are hacked, huge classified data and information will be lost to un-authorized persons globally.

This thesis work focused on Wireless Local Area Networks (WLANs). It examined whether there are vulnerabilities in the IEEE 802.11 security protocols of WLANs. If there are vulnerabilities, it further examined whether the vulnerabilities can be used to hack into a WLAN.

The IEEE 802.11 standard specified three types of security protocols for WLANs: Wired Equivalent Privacy (WEP), Wi-Fi Protected Access (WPA), and Wi-Fi Protected Access 2 (WPA2). Hence, this thesis work focused on discovery vulnerabilities in WEP, WPA, and WPA-2 through experiments.

A laboratory consisting of two laptops with wireless cards, a wireless access point, and an authentication server was set up to probe into the security protocols. A software called BackTrack 5 was installed on one of the laptops. The software was used to launch various attacks in an attempt to discover vulnerabilities and to retrieve the secret keys of WEP, WPA, and WPA-2 networks.

The expectation of this thesis work is to discover a number of vulnerabilities as possible in the IEEE 802.11 Security Protocols of WLANs. Secondly, it is expected to use these vulnerabilities to successfully hack into WLANs.

Table of Content

Chapter 1 Introduction to the Thesis Work

Chapter 2 Literature Review on WLAN Security Protocols

Chapter 3 Experimentation and Thesis Survey

Chapter 5 Conclusion, Recommendation, and Areas for further Research

TABLE OF FIGURES

TABLES

CHAPTER 1

1.1 Introduction

The edge of the internet is increasingly becoming wireless (Tan, 2011). Wireless Local Area Network (WLAN) is the best way to improve data connectivity without the need to worry about wires (Gambiraopet, 2010).

Wireless communication gives organizations and users many benefits, such as increased portability, flexibility, and productivity (Jaiaree, 2003). For example, WLAN devices allow users to move their laptops from place to place within their office without the need for wires and without losing network connectivity (Jaiaree, 2003).

There are numerous applications of WLANs in industries, manufacturing firms, medicine, agriculture, e-commerce, military combat, government sectors, and disaster recovery programs. For example, WLANs are used to transfer images and conversations from an accident scene to the base hospital. Doctors and nurses can use their Personal Digital Accessories (PDAs), mobile phones, and tablets to access patients' database via WLANs without having to move to a fixed computer point to access such information. In military operations, secured WLANs are used to control missiles, fighter jets and the ground crew (Bradford, 2006). During natural disasters like hurricane Katrina, Tsunami, earthquakes, and terrorism, emergency WLANs are setup to allow disaster victims to make phone calls, send emails, or use the internet to contact loved ones. Regardless of the benefits of the WLANs, there are several serious problems that should be addressed prior to deploying a WLAN to supplement a wired system (Jaiaree, 2003).

Firstly, frequency allocation for WLANs presents a problem. This is because most spread-spectrum transmissions for WLANs operate in the non-licensed frequency ranges. This allows for frequency overlaps that causes interferences. Also, other products, like microwave devices, transmit energy in the same spectrum that can potentially induce some level of interference (Jaiaree, 2003).

Secondly, WLANs typically offer lower quality and throughput than wired LANs (Russell, 2002). The main reason for this disadvantage is due to limitations in radio transmission (only 1- 11Mbit/s for 802.11b, 54Mbit/s for 802.11a & 802.11g WLANs). WLANs may also experience high error rates due to interference and longer delays due to multipath propagation (Jaiaree, 2003).

The most critical concern for WLANs is security (Memon et al, 2010). As WLANs transmit their data over open air interfaces, they become susceptible to attacks much easier than with a wired network (Jaiaree, 2003). With a wired network, we can know if any unauthorized person is using the network. This is because to execute an attack, the intruder has to physically connect to the wired network and it is much easy to find that attacker (Gambiraopet, 2010). Whereas in a WLAN, it is possible for the attacker to connect to the network without being in physical contact with it (Park and Dicoi, 2003).

This thesis work discussed the security protocols of WLANs. Each security protocol was examined to identify if there were any vulnerabilities that could be exploited to hack into a Wireless Local Area Network.

1.2 Motivation for this Thesis work

A number of factors stimulated the research undertaken in this thesis. They include the following:

1.2.1 The Escalating Growth of WLANs

The production and use of WLANs in the near future is inevitable. For example, in Ghana, WLANs are found in almost every modern office, shopping malls, and airports. The wide use of WLANs attest to the fact that WLANs are destined to be a promising technology. It is therefore crucial to address the security issues of these networks. This led to the next motivation factor for this thesis work.

1.2.2 The Security Risks Posed by WLANs

Every environment is susceptible to attacks and WLANs are no exception (Abdullah, 2006). The problem is compounded by the multitude of freely available WLAN hacking tools on the internet. This can lead to revenue loss for the WLAN manufacturing companies as many users may decide to abandon the use of WLANs.

WLANs security threats, if not addressed, could have adverse effects on the adequate use of WLANs. This realization led to the next motivating factor.

1.2.3 Addressing WLANs Security Attacks

There exists an urgent need to implement necessary measures to mitigate WLAN security attacks. For instance, the Ghana National Security recently announced the establishment of a national data center where government information will be electronically stored (Joy FM, February 1, 2013). A wireless breach in such a vital environment can be very catastrophic to the government and the people of Ghana.

Also most telecommunication companies and banks in Ghana have wireless networks installed as extensions to their wired infrastructure. If adequate defensive mechanisms are not put in place, an intruder who gets access to the wireless network could automatically gain access to the wired infrastructure. Such an attacker could launch denial of service attack, man-in-the-middle attacks, erase network configurations and steal sensitive data. These attacks can collapse a firm.

In a summary, as Ghana and many other developing countries go electronic, it is very important to educate people about the dangers of wireless networks and to suggest mechanisms to defend against wireless attacks.

1.3 Problem Statement

One can imagine the impact of a vulnerable wireless network to businesses and the society. A vulnerable WLAN could provide attackers with the ability to passively obtain confidential network data and leave no trace of the attack. In addition, they could provide attackers with a backdoor into a network. This can lead to attacks on machines elsewhere on the wired LAN. It is therefore important to identify vulnerabilities in the IEEE 802.11 security protocols of WLANs and to make recommendations to fixing the flaws.

1.4 Thesis Boundary

The study is limited to Wireless Local Area Networks (WLANs). It focuses on IEEE 802.11 a, b, g and i wireless standards. It is centered on wireless infrastructure mode. In addition, software tools running on Windows operating systems were used. Hence the study may not be extended beyond these boundaries.

1.5 Research Questions

I. What are the vulnerabilities in the IEEE 802.11 Security Protocols of WLANs**?**

II. Can WEP, WPA/ WPA2-PSK, and WPA/WPA2-EAP WLAN security protocols be hacked?

1.6 Methodology

The methodology in this thesis consists of literature studies and experiments.

The literature studies consist of published articles of researchers in the fields of wireless local area networks, network security and cryptography. Most of the articles were found on the internet and care was taken to ensure that they are true and have been published.

Secondly, experiments were carried out in an attempt to discover vulnerabilities in the security protocols of WLANs. All experiments were carried out in a laboratory environment. The syntaxes for carrying out the attacks are also provided.

The results from the experiments were used to answer the research questions.

1.7 Definition of Terms

Most of the terms used in this thesis work are explained in the context of the development of the literature review. Below are the explanation of some other terms used:

a. Security

Security is the capability to defend against intrusion (Rackley, 2007). The most common security requirements of WLANs include Authentication, Confidentiality, Access Control, Integrity, and Availability (Jaiaree, 2003; Abrahamsson and Wessman, 2004; Rackley, 2007; Bradra and Hecker, 2008).

b. Authentication

Authentication is the process of verifying that users who attempt to gain access to the network have permission to access the network (Rackley, 2007; Ciampa, 2001). WLANs support two modes of authentication (Khan & Khwaja, 2003). They include Open System and Shared Key Authentications (Abdullah, 2006). In Open System Authentication, any wireless client can associate with the AP and gain access to the network without any authentication (Regan, 2003). Shared Key Authentication requires both the wireless client and the AP to have knowledge of a shared secret key (Abdullah, 2006).

c. Confidentiality

Confidentiality is the process of protecting the transmitted information from unauthorized persons (Jaiaree, 2003). Confidentiality ensures that communication can only be read by authorized persons (Rajib et al, 2008).

d. Access Control

Access Control allows access to the network resources only to those devices and users who have been authenticated successfully (Rackley, 2007).

e. Integrity

Integrity is the ability to ensure that the information transmitted within the wireless network is an accurate and un-modified representation of the original information (Jaiaree, 2003; Rackley, 2007).

f. Availability

Availability is the ability to ensure that the wireless network and its resources are readily accessible to the authorized devices and users whenever needed (Jaiaree, 2003; Bradra and Hecker, 2008).

g. **Vulnerability**

Vyneke and Paggen in 2007 defined Vulnerability as a system or protocol weakness (usually not on purpose) that allows the security of the system or protocol to be compromised. For the purpose of this thesis work, we will consider vulnerability and flaw to mean the same thing.

1.8 Organization of the Thesis work

This work is organized into five chapters:

The first chapter is the introduction to the thesis. It deals with the general overview of the study, motivation for the research, problem statement, thesis boundary, research questions, methodology, definition of terms, and organization of the study.

The second chapter is the literature review. It reviews extensive researches that have been done around the security protocols of WLANs. The first part deals with the literature about the architecture of WEP Protocol. The second and third parts deal with literature on the architecture of WPA/WPA2-PSK and WPA/WPA2-EAP Protocols respectively.

The third chapter is the experiments. It describes the step by step experimental procedure that was used to discover the vulnerabilities in the security protocols of WLANs. It also provides the procedure that was used to successfully hack into these security protocols. Finally, it provides data on 1,271 Access Points that were surveyed in Ghana to discover the security protocol that have been configured on them.

The fourth chapter is on analysis of the experiments and survey. It provides a detailed analysis and significance of the outcome of the experiments.

The fifth chapter is on findings, conclusion, recommendations, and areas of future researches.

CHAPTER 2

LITERATURE REVIEW ON IEEE 802.11 SECURITY PROTOCOLS

2.1 WLAN Security Protocols

The IEEE 802.11 security Group defined three different security protocols to protect WLANs from unauthorized intrusion (Raza et al, 2010; Abdullah, 2006). These included Wired Equivalent Privacy (WEP), Wi-Fi Protected Access (WPA), and Wi-Fi Protected Access 2 (WPA-2) (Raza et al, 2010; Abdullah, 2006). WEP is based on Ron's Code4 (RC4) encryption algorithm (Fluhrer et al, 2001). WPA is also based on RC4 but uses a Temporal Key Integrity Protocol (TKIP) to hash the Initialization Vectors which used to be sent in clear text when using WEP (Beck & Tews, 2009; Arbaugh, 2001). WPA-2 uses the Advanced Encryption Standard (AES) algorithm (Rackley, 2007), and uses a Counter Mode with Cipher Block Chaining Message Authentication Code Protocol (CCMP) to hash the Key (Raza et al, 2010; Abdullah, 2006).

WPA and WPA-2 supports both Enterprise Edition and Personal Edition (Raza et al, 2010; Abdullah, 2006). A RADIUS server or 802.11x Authentication (EAP) is used to support the enterprise edition whiles a Pre-Shared Key (PSK) is used to support the personal edition (Abdullah, 2006) as shown in figure 1.

Figure 1: The supporting security protocols on an 802.11 Access Point

WPA is forward compatible with WPA-2 whereas WPA-2 is backward compatible with WPA (Rackley, 2007; Nwabude, 2008).

2.2 IEEE 802.11 WEP Architecture

According to Nwabude (2008), Jaiaree (2003) and Rackley (2007), WEP is the first encryption scheme made available to Wi-Fi in 1999 (Jaiaree, 2003).

WEP uses RC4 encryption (Raza et al, 2010; Tanzella, 2003). RC4 is a symmetric stream cipher. This means that both the transmitter and receiver use the same key to encrypt and decrypt every data (Khan & Khwaja, 2003; Raza et al, 2010), The RC4 Algorithm requires an input key of size 64-bit or 128-bit (Chandra et al, 2009; Vivek, 2011).WEP uses a 40-bit or 104-bit key plus a 24bit cryptographic salt called an Initialization Vector (IV) as a seed to the RC4 algorithm (Chandra et al, 2009; Vivek, 2011) as shown in figure 2.

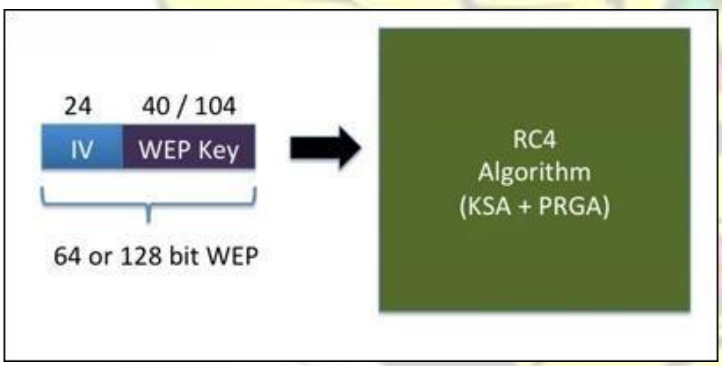


Figure 2: A 64 or 128 bit WEP key as a seed to the RC4 Algorithm

The WEP field is located within the frame body of the WLAN packet (Vivek, 2011) as shown in figure 3.

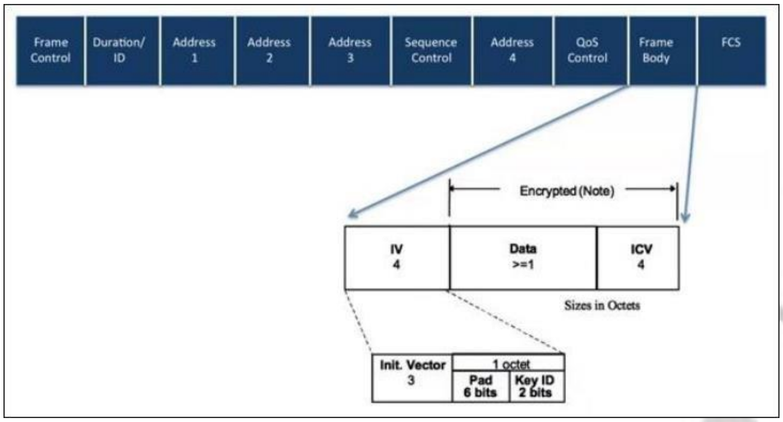


Figure 3: The WEP field located within the frame body of the WLAN Packet

The WEP field consists of three main parts: a 4-byte IV, the encrypted data, and a 4-byte Integrity Check Value (ICV) to guarantee the integrity of the data (Chandra et al, 2009; IEEE std 802.11, 2012).

2.2.1 Initialization Vector (IV)

Since the WEP Key is of a fixed length and is static, it would output the same keystream bytes when it is input into the RC4 algorithm as shown in figure 4.

Figure 4: Static WEP key put in RC4 outputs static keystream bytes

In order to randomize the output keystream bytes, the IEEE 802.11 security group introduced an Initialization Vector (Vivek, 2011). The IV is a 24-bit randomized value. The IV is then pre-pended to the WEP key and fed as input to the RC4 Algorithm so that the output keystream bytes will be randomized. (Vivek, 2011) as shown in figure 5.

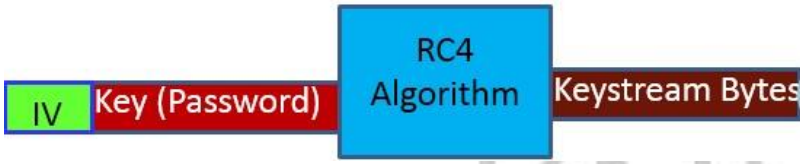


Figure 5: Static WEP key prepended with a 24-bit IV put in RC4 outputs randomized keystream bytes

The IV field consists of 3 bytes: 6 bits of padding and 2 bits of Key ID (IEEE std 802.11, 2012) as shown in figure 3. The Key ID field denotes the different WEP keys that can be configured on an AP at the same time (Vivek, 2011). Every AP allows for a maximum of four different WEP keys which are distinguished by the Key ID bits field 00, 01, 10, or 11 (Chandra et al, 2009) as shown in figure 6.

Figure 6: Four different WEP keys can be configured on an AP at the same time

2.2.2 The RC4 Algorithm

The RC4 algorithm consists of two main parts called the Key Scheduling Algorithm (KSA) and the Pseudo-Random Generation Algorithm (PRGA) (Chandra et al, 2009). The following subsections discusses into detail how the KSA and PRGA algorithms are generated:

A. KEY SCHEDULING ALGORITHM (KSA)

a. The KSA first establishes a 256-byte array with a permutation of the numbers 0 to 255 in which all the elements are preset to 0:

   S[box] : S[0]=0 S[1]=0 S[2]=0 S[3]=0 S[4]=0 … S[254]=0 S[255]=0

   The procedure is described with the below pseudorandom code:

   For i=0 to 255, S[i]=0.

b. The S[box] is then re-initialized to S[i]=i to give

   S[box] : S[0]=0 S[1]=1 S[2]=2 S[3]=3 S[4]=4 … S[254]=254 S[255]=255.

   The pseudorandom code for the above sequence is given by

   For i= 0 to 255, S[i]= i.

Figure 7 shows an example of a 256-byte array that have been initialized to S[i]= i:

0	26	52	78	104	130	156	182	208	234
1	27	53	79	105	131	157	183	209	235
2	28	54	80	106	132	158	184	210	236
3	29	55	81	107	133	159	185	211	237
4	30	56	82	108	134	160	186	212	238
5	31	57	83	109	135	161	187	213	239
6	32	58	84	110	136	162	188	214	240
7	33	59	85	111	137	163	189	215	241
8	34	60	86	112	138	164	190	216	242
9	35	61	87	113	139	165	191	217	243
10	36	62	88	114	140	166	192	218	244
11	37	63	89	115	141	167	193	219	245
12	38	64	90	116	142	168	194	220	246
13	39	65	91	117	143	169	195	221	247
14	40	66	92	118	144	170	196	222	248
15	41	67	93	119	145	171	197	223	249
16	42	68	94	120	146	172	198	224	250

17	43	69	95	121	147	173	199	225	251
18	44	70	96	122	148	174	200	226	252
19	45	71	97	123	149	175	201	227	253
20	46	72	98	124	150	176	202	228	254
21	47	73	99	125	151	177	203	229	255
22	48	74	100	126	152	178	204	230	
23	49	75	101	127	153	179	205	231	
24	50	76	102	128	154	180	206	232	
25	51	77	103	129	155	181	207	233	

Figure 7: A 256-byte array initialized to S[i]= i

c. The 64 or 128 bit WEP key (IV plus WEP key) is converted into their ASCII characters (see Appendix A) and repeatedly used to fill a 256-byte array (Chandra et al, 2009). For example, if the WEP key entered by the user is "KEY" (here we exclude the IV for simplicity sake), and is used to repeatedly fill a 256-byte K[box] array.

   K[box] : K[0]=K K[1]=E K[2]=Y K[3]=K K[4]=E K[5]=Y K[6]=K … K[254]=Y K[255]=K

Figure 8 shows a repeated filling of a 256-byte array with the WEP Key. In this case, the key is "KEY" without any IV for simplicity sake.

K	Y	E	K	Y	E	K	Y	E	K
E	K	Y	E	K	Y	E	K	Y	E
Y	E	K	Y	E	K	Y	E	K	Y
K	Y	E	K	Y	E	K	Y	E	K
E	K	Y	E	K	Y	E	K	Y	E
Y	E	K	Y	E	K	Y	E	K	Y
K	Y	E	K	Y	E	K	Y	E	K
E	K	Y	E	K	Y	E	K	Y	E
Y	E	K	Y	E	K	Y	E	K	Y
K	Y	E	K	Y	E	K	Y	E	K
E	K	Y	E	K	Y	E	K	Y	E
Y	E	K	Y	E	K	Y	E	K	Y
K	Y	E	K	Y	E	K	Y	E	K
E	K	Y	E	K	Y	E	K	Y	E
Y	E	K	Y	E	K	Y	E	K	Y
K	Y	E	K	Y	E	K	Y	E	K
E	K	Y	E	K	Y	E	K	Y	E

Y	E	K	Y	E	K	Y	E	K	Y
K	Y	E	K	Y	E	K	Y	E	K
E	K	Y	E	K	Y	E	K	Y	E
Y	E	K	Y	E	K	Y	E	K	Y
K	Y	E	K	Y	E	K	Y	E	K
E	K	Y	E	K	Y	E	K	Y	
Y	E	K	Y	E	K	Y	E	K	
K	Y	E	K	Y	E	K	Y	E	
E	K	Y	E	K	Y	E	K	Y	

Figure 8: A repeated filling of a 256-byte array with the WEP Key “ KEY ”

d. Next, the key is converted into their ASCII characters: The K[box] becomes

K[box]: K[0]=75 K[1]=69 K[2]=89 K[3]=75 K[4]=69 K[5]=89 … K[254]=89 K[255]=75

Figure 9 shows the content of figure 8 when converted into their ASCII characters:

75	89	69	75	89	69	75	89	69	75
69	75	89	69	75	89	69	75	89	69
89	69	75	89	69	75	89	69	75	89
75	89	69	75	89	69	75	89	69	75
69	75	89	69	75	89	69	75	89	69
89	69	75	89	69	75	89	69	75	89
75	89	69	75	89	69	75	89	69	75
69	75	89	69	75	89	69	75	89	69
89	69	75	89	69	75	89	69	75	89
75	89	69	75	89	69	75	89	69	75
69	75	89	69	75	89	69	75	89	69
89	69	75	89	69	75	89	69	75	89
75	89	69	75	89	69	75	89	69	75
69	75	89	69	75	89	69	75	89	69
89	69	75	89	69	75	89	69	75	89
75	89	69	75	89	69	75	89	69	75
69	75	89	69	75	89	69	75	89	69

89	69	75	89	69	75	89	69	75	89
75	89	69	75	89	69	75	89	69	75
69	75	89	69	75	89	69	75	89	69
89	69	75	89	69	75	89	69	75	89
75	89	69	75	89	69	75	89	69	75
69	75	89	69	75	89	69	75	89	
89	69	75	89	69	75	89	69	75	
75	89	69	75	89	69	75	89	69	
69	75	89	69	75	89	69	75	89	

Figure 9 The WEP key “ KEY ” converted into their ASCII characters

e. The S[box] and the K[box] are acted upon using the below pseudorandom algorithm:

i= j= 0;

For i= 0 to 255 do j= (j +

S[i] + K[i]) mod 256;

Swap S[i] and S[j];

End; // j is a single byte value and any overflow in the addition is ignored.

The algorithm which involves 256 iteration is very simple to illustrate:

<u>Before the first iteration</u>

S[box] : S[0]=0 S[1]=1 S[2]=2 S[3]=3 S[4]=4 … S[254]=254 S[255]=255.

K[box]: K[0]=75 K[1]=69 K[2]=89 K[3]=75 K[4]=69 K[5]=89 … K[254]=89 K[255]=75

For i=0, j= (previous j + S[i] + K[i]) mod 256.

Thus j= (0 +0 +75) mod 256 = 75 mod 256 = 75.

Now swapping the content of positions S[i] and S[j],

The content of S[0] which used to be 0 now becomes 75 whilst the content of S[75] which used to be 75 now becomes 0.

After the first iteration

S[box] : S[0]=75 S[1]=1 S[2]=2 S[3]=3 S[4]=4 … S[75]=0 … S[254]=254 S[255]=255.

Before the second iteration

S[box] : S[0]=75 S[1]=1 S[2]=2 S[3]=3 S[4]=4 … S[75]=0 … S[254]=254 S[255]=255.

K[box]: K[0]=75 K[1]=69 K[2]=89 K[3]=75 K[4]=69 K[5]=89 … K[254]=89 K[255]=75

For i=1, j= (previous j + S[i] + K[i]) mod 256.

Thus j= (75 +1 +69) mod 256 = 145 mod 256 = 145. Now

swapping the content of positions S[i] and S[j],

The content of S[1] which used to be 1 now becomes 145 whilst the content of S[145] which used to be 145 now becomes 1.

After the second iteration

S[box] : S[0]=75 S[1]=145 S[2]=2 S[3]=3 S[4]=4 … S[75]=0 … S[145]=1 … S[254]=254 S[255]=255.

Before the third iteration

S[box] : S[0]=75 S[1]=145 S[2]=2 S[3]=3 S[4]=4 … S[75]=0 … S[145]=1 … S[254]=254 S[255]=255.

K[box]: K[0]=75 K[1]=69 K[2]=89 K[3]=75 K[4]=69 K[5]=89 … K[254]=89 K[255]=75

For i=2, j= (previous j + S[i] + K[i]) mod 256.

Thus j= (145 +2 +89) mod 256 = 236 mod 256 = 236.

Now swapping the content of positions S[i] and S[j],

The content of S[2] which used to be 2 now becomes 236 whilst the content of S[236] which used to be 236 now becomes 2.

<u>After the third iteration</u>

S[box] : S[0]=75 S[1]=145 S[2]=236 S[3]=3 S[4]=4 … S[75]=0 … S[145]=1 … S[236]=2 … S[254]=254 S[255]=255.

The position swaps continue to run up to i=255.

In a summary, j= (j+S[i]+K[i]) mod 256 which determines the position to swap within the S[box] array is shown in figure 10:

75	144	107	254	47	246	117	146	69	176
145	246	249	146	227	210	87	148	111	224
236	87	122	59	146	161	78	145	140	37
58	205	246	215	86	107	56	163	164	93
131	54	135	110	13	74	29	168	209	144
225	154	11	26	191	28	23	168	241	216
50	19	138	185	134	233	4	189	12	19
126	127	30	83	64	203	236	197	60	73
223	230	165	2	245	160	233	200	95	148
51	98	39	164	191	112	217	224	125	210
130	209	190	65	124	85	196	235	176	11
230	59	72	243	52	45	196	241	214	89
61	186	205	152	1	0	183	12	247	154
143	44	103	56	193	232	165	26	45	214
246	153	244	237	124	195	168	35	86	39
80	27	124	149	76	153	158	65	122	107
165	144	25	56	15	132	143	82	179	170
15	0	169	240	205	98	149	94	223	254
108	133	52	155	160	59	142	127	6	69
196	253	212	65	102	41	130	147	66	135
49	112	103	252	39	10	139	162	113	222
145	248	245	170	253	230	135	198	155	40
236	115	152	83	198	215	126	221	218	
92	233	46	17	138	187	138	239	12	
191	116	191	194	99	154	137	22	57	
29	242	101	110	47	142	131	48	123	

Figure 10: The position swap generator j for all i=0 to 255

After the complete scrambling or position swapping of the entire 256-byte S[box] array, a new 256-byte array is generated which is called the KSA as shown in figure 11:

142	42	48	158	245	175	193	161	169	218
6	41	249	146	227	181	44	15	71	176
236	56	122	37	79	106	171	40	89	59
58	33	27	215	53	145	94	22	136	149
162	54	159	7	143	152	205	99	25	195
225	154	13	85	76	125	92	38	117	240
226	196	138	185	134	20	202	189	220	32
126	127	148	140	36	203	84	197	16	104
223	11	90	222	139	180	211	200	121	168

51	167	157	164	228	179	69	224	45	144
66	209	190	103	67	4	19	235	151	150
57	163	72	46	70	253	166	241	102	243
231	186	111	74	86	174	21	61	247	31
18	47	188	135	156	172	217	83	141	219
246	255	105	237	114	238	250	98	251	118
120	93	124	23	24	184	254	97	5	39
165	26	29	199	82	182	232	137	192	183
77	75	35	216	64	201	14	91	160	78
108	133	107	155	8	9	96	3	50	208
95	204	212	88	28	173	55	252	244	177
49	0	65	147	12	130	234	131	109	52
1	129	213	128	60	34	87	110	43	153
206	207	63	85	178	81	198	221	73	
132	233	112	80	62	187	113	239	214	
191	116	119	123	170	10	30	2	230	
68	242	101	17	248	229	210	115	194	

Figure 11: The generated KSA

The generated KSA now becomes an input to the Pseudo-Random Generation Algorithm (PRGA).

B. PSEUDO-RANDOM GENERATION ALGORITHM (PRGA)

Once the KSA has been completed, the next phase in the RC4 is the PRGA (Chandra et al, 2009; Vivek, 2011). This phase involves more swapping of bytes in the KSA and generates one pseudorandom byte per iteration (Chandra et al, 2009). Each pseudorandom byte is then XORed with each byte of the plaintext to encrypt it (Vivek, 2011).

The PRGA is given by the below pseudocode (Chandra et al, 2009):

i= j= 0; i= (i + 1) mod 256;

j= (j + S[i]) mod 256; Swap

S[i] and S[j]; k= S[S[i] +

S[j]] mod 256;

// k is the generated pseudorandom byte to be XORed with each byte of the plaintext.

We use i (from 0 to 255) and the KSA as shown in figure 12 to implement the PRGA algorithm.

0	26	52	78	104	130	156	182	208	234
1	27	53	79	105	131	157	183	209	235
2	28	54	80	106	132	158	184	210	236
3	29	55	81	107	133	159	185	211	237
4	30	56	82	108	134	160	186	212	238
5	31	57	83	109	135	161	187	213	239
6	32	58	84	110	136	162	188	214	240
7	33	59	85	111	137	163	189	215	241
8	34	60	86	112	138	164	190	216	242
9	35	61	87	113	139	165	191	217	243
10	36	62	88	114	140	166	192	218	244
11	37	63	89	115	141	167	193	219	245
12	38	64	90	116	142	168	194	220	246
13	39	65	91	117	143	169	195	221	247
14	40	66	92	118	144	170	196	222	248
15	41	67	93	119	145	171	197	223	249
16	42	68	94	120	146	172	198	224	250
17	43	69	95	121	147	173	199	225	251
18	44	70	96	122	148	174	200	226	252
19	45	71	97	123	149	175	201	227	253
20	46	72	98	124	150	176	202	228	254
21	47	73	99	125	151	177	203	229	255
22	48	74	100	126	152	178	204	230	
23	49	75	101	127	153	179	205	231	
24	50	76	102	128	154	180	206	232	
25	51	77	103	129	155	181	207	233	

I = 0 to 255

142	42	48	158	245	175	193	161	169	218
6	41	249	146	227	181	44	15	71	176
236	56	122	37	79	106	171	40	89	59
58	33	27	215	53	145	94	22	136	149
162	54	159	7	143	152	205	99	25	195
225	154	13	85	76	125	92	38	117	240
226	196	138	185	134	20	202	189	220	32
126	127	148	140	36	203	84	197	16	104
223	11	90	222	139	180	211	200	121	168
51	167	157	164	228	179	69	224	45	144
66	209	190	103	67	4	19	235	151	150
57	163	72	46	70	253	166	241	102	243
231	186	111	74	86	174	21	61	247	31
18	47	188	135	156	172	217	83	141	219
246	255	105	237	114	238	250	98	251	118
120	93	124	23	24	184	254	97	5	39
165	26	29	199	82	182	232	137	192	183
77	75	35	216	64	201	14	91	160	78
108	133	107	155	8	9	96	3	50	208
95	204	212	88	28	173	55	252	244	177
49	0	65	147	12	130	234	131	109	52
1	129	213	128	60	34	87	110	43	153
206	207	63	85	178	81	198	221	73	
132	233	112	80	62	187	113	239	214	
191	116	119	123	170	10	30	2	230	
68	242	101	17	248	229	210	115	194	

KSA = S[i]

Figure 12: i from 0 to 255 and the KSA used as seed into the PRGA

First keystream byte generation i= (previous i + 1) mod 256 = (0 + 1) mod 256 = 1 mod 256 = 1; j= (previous j + S[i]) mod 256 = (0 + 6) mod 256 = 6 mod 256 = 6;

Swapping the content of positions S[i] and S[j],

The content of S[1] which used to be 6 now becomes 226 whilst the content of S[6] which used to be 226 now becomes 6.

Now generating the first keystream byte (k):

k = S[S[i] + S[j]] mod 256 = S[226 + 6] mod 256 = S[232] mod 256.

But S[232]= 230. Thus k= 230 mod 256 = 230.

Thus the first keystream byte is 230 or 11100110 (in base 2) and will be XORed with the first plaintext byte to encrypt it.

<u>Second keystream byte generation</u> i= (previous i + 1) mod 256 = (1 + 1)

mod 256 = 2 mod 256 = 2; j= (previous j + S[i]) mod 256 = (6 + 236) mod

256 = 242 mod 256 = 242;

Swapping the content of positions S[i] and S[j],

The content of S[2] which used to be 236 now becomes 168 whilst the content of S[242] which used to be 168 now becomes 236.

Now generating the second keystream byte (k):

k = S[S[i] + S[j]] mod 256 = S[168 + 236] mod 256 = S[404] mod 256.

But 404 mod 256 = 148,

Hence S[404] mod 256 = S[148] mod 256.

But S[148]= 9. Thus k= 9 mod 256 = 9.

Thus the second keystream byte is 9 or 00001001 (base 2) and will be XORed with the second plaintext byte to encrypt it.

In all, 256 keystream bytes are generated. Note that each time a word of the keystream is generated the internal state of RC4 is updated (Hulton, 2002).

In a summary, j= (previous j + S[i]) mod 256 which determines the position in the KSA to swap is shown in figure 13:

6	154	239	171	226	83	107	23	60	45
242	210	105	208	49	189	22	63	149	104
44	243	132	167	102	78	116	85	29	253
206	41	35	174	245	230	65	184	54	192
175	195	48	3	65	99	157	222	171	176
145	135	186	188	199	119	103	155	135	208
15	6	78	72	235	66	187	96	151	56
238	17	168	38	118	246	142	40	16	224
33	184	69	202	90	169	211	8	61	112
99	137	3	49	157	173	230	243	212	6
156	44	75	95	227	170	140	228	58	249
131	230	186	169	57	88	161	33	49	24
149	21	118	48	213	4	122	116	190	243
139	20	223	29	71	242	116	214	185	105
3	113	91	52	95	170	114	55	190	144
168	139	120	251	177	96	90	192	126	71
245	214	155	211	241	41	104	27	30	149
97	91	6	110	249	50	200	30	80	101
192	39	218	198	21	223	255	26	68	22
241	39	27	89	33	97	233	157	177	74
242	168	240	217	93	131	64	11	220	227
192	119	47	46	15	212	6	232	37	
68	96	159	126	77	143	119	215	251	
3	212	22	249	247	153	149	217	225	
71	198	123	10	239	126	103	76	163	
113	246	25	255	158	63	8	245	125	

Figure 13:The position swap generator j for the KSA

After the complete swapping of the positions in the KSA for all i=0 to 255, the resulting KSA is given by figure 14:

142	10	240	138	245	175	57	161	169	218
226	89	227	146	249	181	44	15	71	176
168	144	106	37	79	122	171	11	41	59
157	93	203	215	53	145	94	22	136	149
2	83	155	7	143	152	205	72	233	223
55	125	99	85	76	154	92	38	117	48
196	184	158	185	134	20	202	189	26	32
120	88	0	140	36	167	84	197	16	95
195	40	35	222	139	180	211	200	121	49
127	27	132	164	255	33	69	224	45	56
128	58	112	103	67	4	19	1	151	150
193	73	13	46	70	253	166	241	102	165
173	108	114	74	86	174	148	61	247	242
179	204	188	75	156	172	217	54	141	219
133	68	105	237	111	238	250	98	251	118
21	18	124	23	129	6	254	97	5	39
243	220	206	199	82	182	232	116	192	183
51	135	90	216	64	201	14	91	160	78
235	236	107	207	8	9	96	3	50	208
104	209	191	77	28	231	225	252	244	177
47	126	65	147	12	130	234	131	109	52
186	24	213	66	60	34	87	110	43	153
29	159	63	85	178	81	198	221	163	
246	25	190	80	62	187	113	239	214	
212	137	119	123	170	42	30	162	230	
228	31	101	17	248	229	210	115	194	

Figure 14: The resulting KSA after 256 iterations

After the complete iteration of the KSA, the first 256 PRAG keystream bytes is generated. The output of the first 125 PRGA is shown in figure 15. It is first shown in decimal notation and then in binary notation:

230	158	184	24	23	186
9	175	247	20	250	95
224	3	109	0	204	237
211	178	163	110	44	190
191	203	138	203	9	124
200	246	139	211	37	187
31	12	68	235	246	245
202	18	9	45	201	213
198	242	60	118	220	52
61	67	88	77	135	
183	193	126	197	203	
9	59	222	13	46	
97	108	160	120	165	
28	89	241	194	250	
201	124	30	116	121	
81	242	162	141	102	
69	31	113	80	238	
164	41	83	169	61	
91	197	168	149	115	
33	44	212	82	139	
143	187	77	140	215	
176	79	149	1	70	
8	168	44	1	107	
201	177	102	120	84	
255	51	201	129	5	

Figure 15: The resulting PRGA after 125 iterations in decimal notations

11100110	10011110	10111000	11000	10111	10111010
1001	10101111	11110111	10100	11111010	1011111
11100000	11	1101101	0	11001100	11101101
11010011	10110010	10100011	1101110	101100	10111110
10111111	11001011	10001010	11001011	1001	1111100
11001000	11110110	10001011	11010011	100101	10111011
11111	1100	1000100	11101011	11110110	11110101
11001010	10010	1001	101101	11001001	11010101
11000110	11110010	111100	1110110	11011100	110100
111101	1000011	1011000	1001101	10000111	
10110111	11000001	1111110	11000101	11001011	
1001	111011	11011110	1101	101110	
1100001	1101100	10100000	1111000	10100101	
11100	1011001	11110001	11000010	11111010	
11001001	1111100	11110	1110100	1111001	
1010001	11110010	10100010	10001101	1100110	
1000101	11111	1110001	1010000	11101110	
10100100	101001	1010011	10101001	111101	
1011011	11000101	10101000	10010101	1110011	
100001	101100	11010100	1010010	10001011	
10001111	10111011	1001101	10001100	11010111	
10110000	1001111	10010101	1	1000110	
1000	10101000	101100	1	1101011	
11001001	10110001	1100110	1111000	1010100	
11111111	110011	11001001	10000001	101	

Figure 15 b: The resulting PRGA after 125 iterations in binary notations

2.2.3 The Integrity Check Value (ICV)

A 32-bit Cyclic Redundancy Checksum (CRC-32) is computed and appended to the data prior to encryption (Chandra et al, 2009; Vivek, 2011; IEEE 802.11, 2012). The ICV which is the computed CRC-32 is to prevent anyone from tempering with the data in transit. This ICV would be exploited to see if there are any vulnerabilities that can be used to hack into the WEP protocol.

The CRC-32 is computed by using a polynomial function called the Generator G(x) which is known to both the transmitter and receiver (Peterson & Brown, 1961; Chaabouni, 2006). G(x) is XORed with the data to yield the 32-bit checksum (Chaabouni, 2006). The size of the checksum is depended on the highest degree of the Generator (Peterson & Brown, 1961). Hence to yield a 32-bit checksum, the generator should necessarily contain a highest degree of 32.

Consider an agreed Generator G(x) between the transmitter and receiver of the form $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{2} + x + 1$, the CRC-32 checksum on a data or message say " Hi " is computed as follows:

1. The message is converted into ASCII (see Appendix A) and then into binary. For example, " Hi " becomes H = 072 = 01001000 (base 2) and i = 105 = 01101001. Hence Hi = 0100100001101001. The below URL was used to convert the text into binary: http://www.roubaixinteractive.com/PlayGround/Binary_Conversion/Binary_To_Text.asp

2. The generator G(x) is also converted into binary. By comparing G(x) with the below polynomial P(x) where k is the highest degree of the polynomial, the value is 1 if the degree of the polynomial exist and is 0 if the degree does not exist:

$P(x) = x^{k} + x^{k-1} + x^{k-2} + \ldots + x^{3} + x^{2} + x + 1.$

$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{2} + x + 1.$

Thus G(x) into binary = 100000100110000010001110110100111

3. Since the highest degree of the G(x) is 32, we append 32 zeroes to the message. Thus the message " Hi " becomes

M(x) = 010010000110100100000000000000000000000000000000

4. M(x) is considered as the dividend and G(x) is the divisor. The mathematical relation, Dividend = (Divisor × Quotient) + Remainder is used to compute the remainder which is the CRC-32.

5. The concept of exclusive OR (XOR) is used in this computation to obtain the remainder. XOR combines two bytes and generates a single byte (Chandra et al, 2009; Vivek, 2011). It compares corresponding bits in each byte. If they are equal, the result is 0; if they differ, the result is 1 (Chandra et al, 2009; Vivek, 2011; Peterson & Brown, 1961). Thus

   0 XOR 0 = 0
   0 XOR 1 = 1
   1 XOR 0 = 1
   1 XOR 1 = 0

6. The remainder (CRC-32) is computed as follows:

```
100000100110000010001110110100111√ 01001000011010010000000000000000000000000000000
XOR                                 100000100110000010001110110100111
---------------------------------------------------------------------------------------
                                    110010100000100110001110110100111
XOR                                 100000100110000010001110110100111
---------------------------------------------------------------------------------------
                                    0100100001101001000000000000000000
XOR                                  100000100110000010001110110100111
---------------------------------------------------------------------------------------
                                     000100101011001010001110110100111000
XOR                                     100000100110000010001110110100111
---------------------------------------------------------------------------------------
                                        000101111111010011111000010011111000
XOR                                        100000100110000010001110110100111
---------------------------------------------------------------------------------------
```

	00111101110001110100110010101111100
XOR	100000100110000010001110110100111
	01110101011111101101111000110110110
XOR	100000100110000010001110110100111
	011010001001101111111011000000100010
XOR	100000100110000010001110110100111
	0101001101010111011000101100001010
XOR	100000100110000010001110110100111
	00100100110011100100101101010110100
XOR	100000100110000010001110110100111
Remainder (CRC-32) =	0001000101011001101000111000100110

Thus the computed ICV or 32-bit Cyclic Redundancy Checksum for the message “ Hi ” is

01000101011001101000111000100110.

Regardless of the size of the data, the output of the ICV is always 32 bits (Abdullah, 2006; Vivek, 2011).

2.2.4 WEP Data Encryption Mechanism

a. The ICV and the data are concatenated into one block as shown in figure 16.

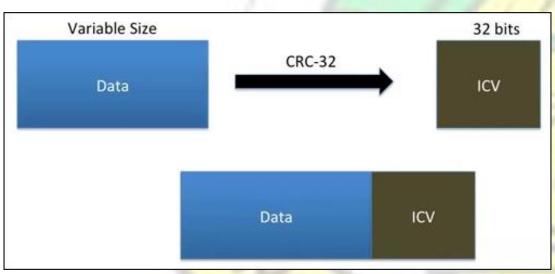
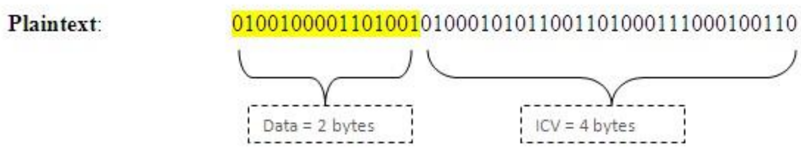


Figure 16: The ICV (32-bits) and the Data concatenated into one block

Thus for example, the data “Hi “and the computed ICV in section 2.2.3 becomes



b. Next we take a Random Keystream of the same size as the concatenated Data and ICV as shown in figure 17.

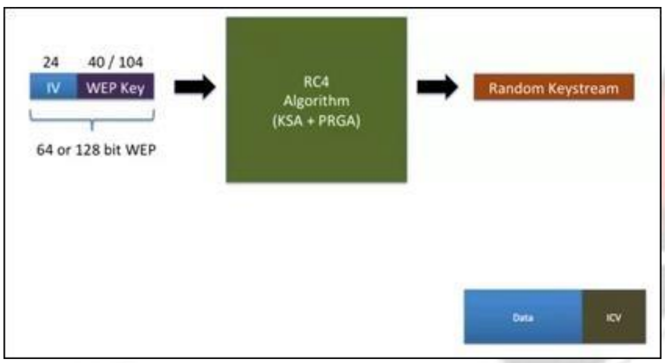
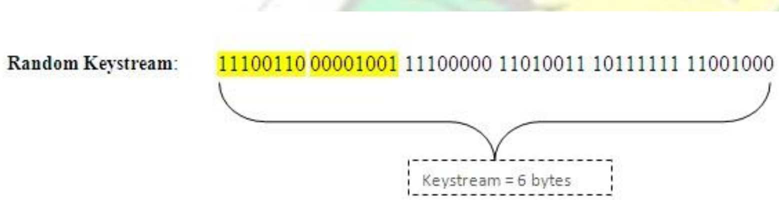


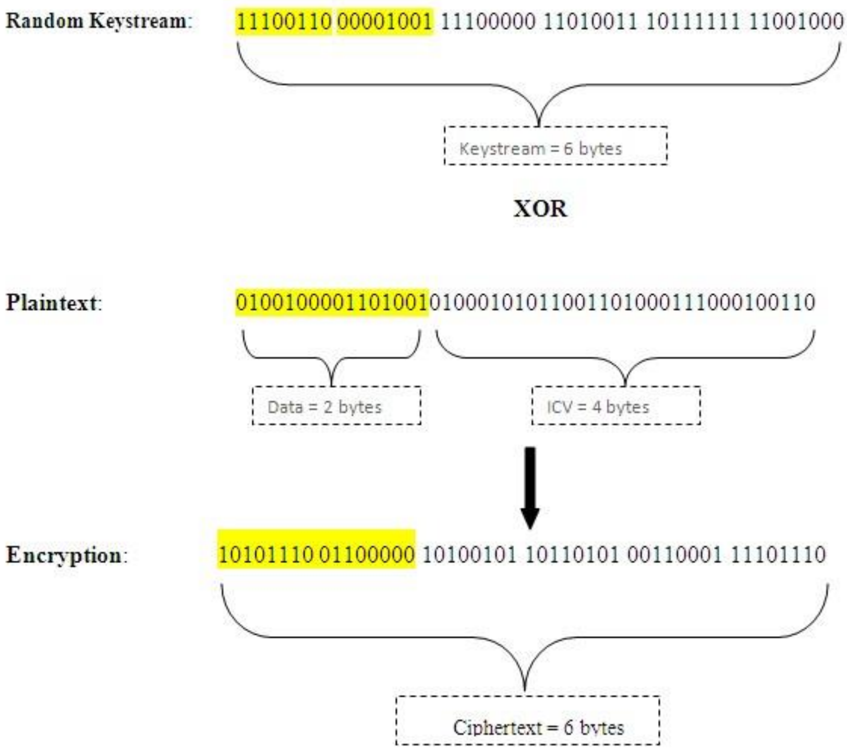
Figure 17: The random keystream byte of the same length as the data and ICV

Thus, for example, we will take the first 6 bytes from the random keystream generated in section 2.2.2 (see figure 15b) as shown below:



c. We perform a XOR (®) operation of the Random Keystream and the Plaintext to obtain the encrypted or Ciphertext as follows:

Encryption: Plaintext ® Random Keystream = Ciphertext **Decryption**:
Ciphertext ® Random Keystream = Plaintext



d. For the purpose of decryption, since the receiving station knows the WEP Key but does not know the IV which the transmitter used, we take the IV field consisting of 3 bytes, 6 bits of padding and 2 bits of Key ID and append it to the Ciphertext as shown in figure 18.

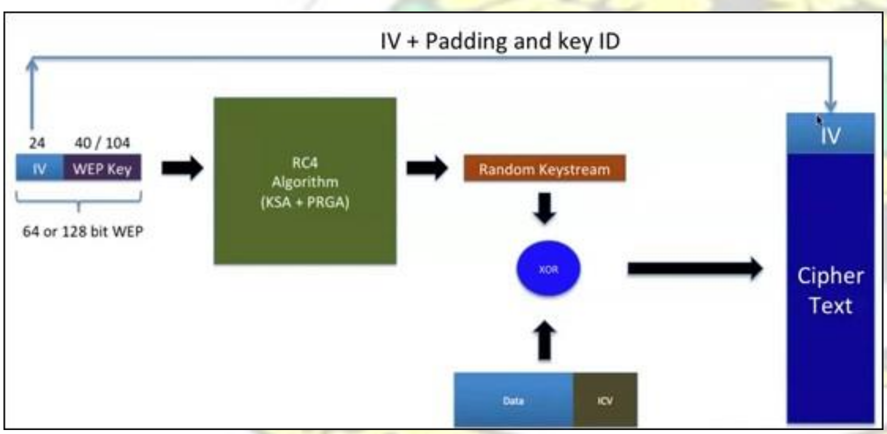


Figure 18: The IV appended to the Ciphertext prior to transmission

e. Finally the whole IEEE 802.11 WLAN frame consisting of the WEP Header is transmitted over the air interface as shown in figure 19.

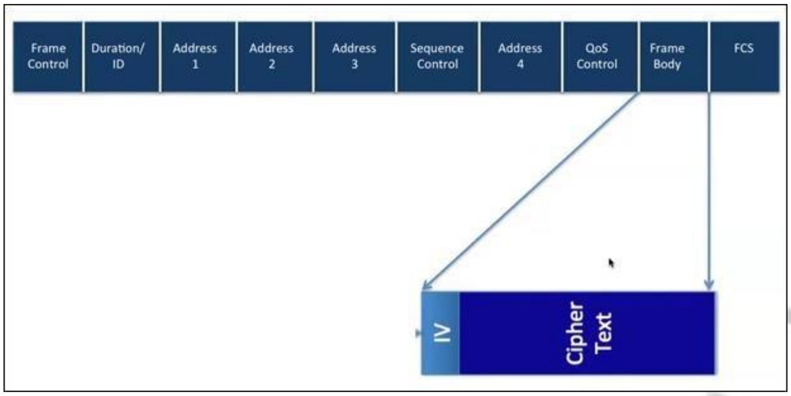


Figure 19: The WLAN packet with embedded WEP Header

Figure 20 shows the location of the IV, key ID, and the encrypted ICV in a WLAN WEP packet captured with Wireshark:

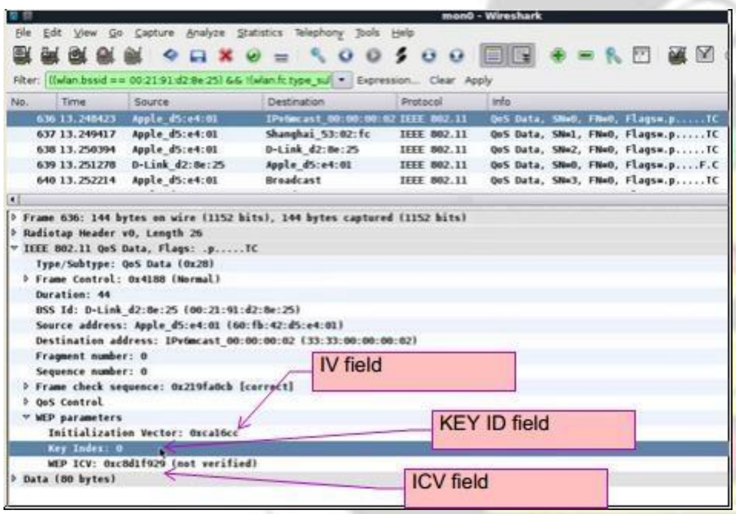


Figure 20: The location of the IV, key ID, and the encrypted ICV in a WLAN packet captured with Wireshark

Figure 21 summarizes the WEP Data Encryption process with a WEP Encapsulation block diagram:

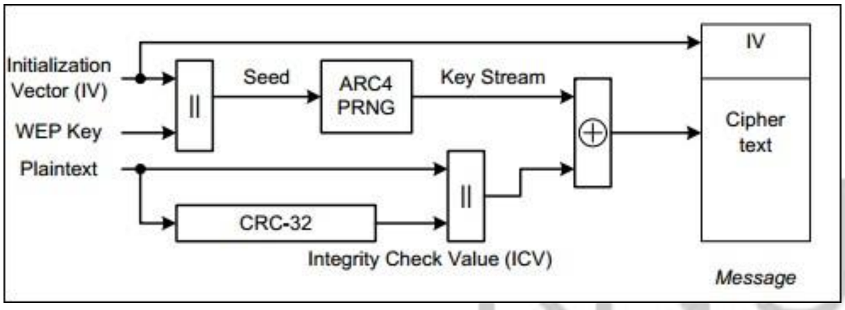


Figure 21: WEP encapsulation block diagram (IEEE 802.11, 2012)

2.2.5 WEP Data Decryption Mechanism

Since the Ciphertext XOR the Random Keystream gives the Plaintext (Chandra et al, 2009; Vivek, 2011; Peterson & Brown, 1961), the receiving station performs the following to decrypt the message:

a. It reads the key ID in the encrypted message as shown figure 20 in order to select the correct WEP key for decryption.

b. It takes the IV field from the encrypted message (see figure 20), and append the WEP Key to it.

c. It passes both the IV and the WEP Key as a seed to the RC4 algorithm (KSA and PRGA) to generate the correct Random Keystream bytes as shown in figure 22.

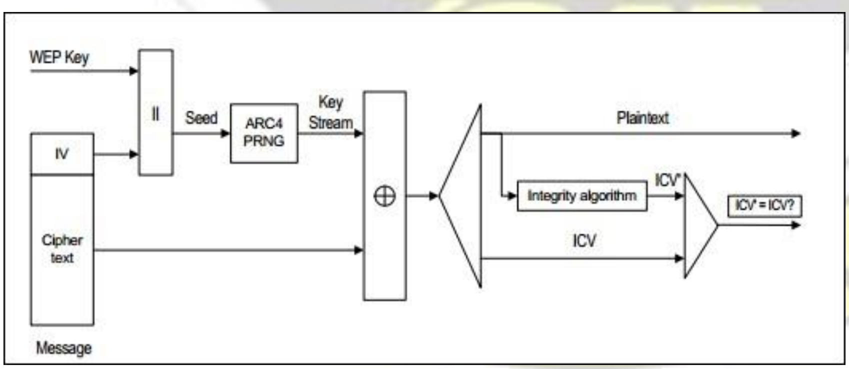
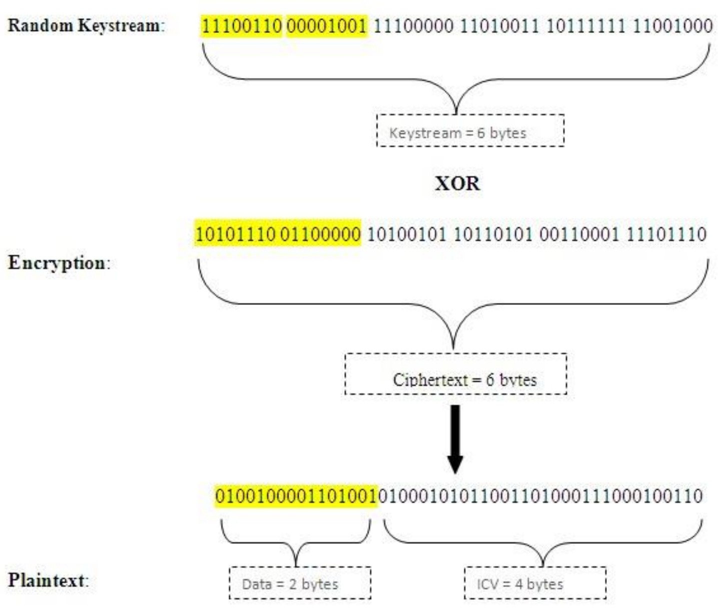


Figure 22 shows WEP decapsulation block diagram (IEEE 802.11, 2012)

d. It takes the Ciphertext and the exact same size of random keystream, and performs a XOR (®) operation to obtain the plaintext.

Decryption: Ciphertext ® Random Keystream = Plaintext



e. The decrypted (plaintext) message now consist of the actual data and an ICV whose position and size (32-bits) is known.

f. The receiving station detaches the ICV, leaving the actual data, and computes an ICV on the actual data using the same generator G(x) (see section 2.2.3).

g. If the computed ICV matches with the transmitted or detached ICV (see figure 20), it means the integrity of the data has not been tempered with during transmission. If there is no match, the data is discarded and the transmitting station is informed to retransmit the data.

2.3 IEEE 802.11 WI-FI PROTECTED ACCESS Pre-Shared Key (WPA/ WPA2-PSK) Architecture

In 2001, the IEEE 802.11i group was tasked to design a new security protocol for the 802.11 family of WLANs (Rackley, 2007; Winget et al, 2007; Sithirasenan, 2004). The IEEE group designed two protocols; one that will require legacy WEP devices to receive firmware or software upgrades (WPA) and the other that required both hardware and firmware changes (WPA-2) (Sithirasenan, 2004). These two protocols were named Temporal Key Integrity Protocol (TKIP) and Counter Mode with CBC MAC Protocol (CCMP) respectively (Wi-Fi Alliance, 2003; Edney & Arbaugh, 2004; Akin, 2005). TKIP was designed based on the existing RC4 architecture whilst CCMP was designed based on the Advanced Encryption Standard (AES) block cipher (Halvorsen & Haugen, 2009).

2.3.1 Architecture OF TKIP

TKIP defines four modifications as patches to WEP:

1. The use of dynamic keys instead of static keys. WPA provides both a pairwise master key (PMK) and pairwise temporal keys (PTK). The PTKs are generated on a per packet basis. Once a PTK is compromised, only the packet encrypted with that PTK can be decrypted; Other PTKs are not compromised (Halvorsen & Haugen, 2009).
2. Provides support for mutual authentication. Both the client and AP authenticate each other through the use of message integrity checking (MIC).

3. Expands the size of the IV space from 24 bits to 48 bits with sequencing rules using TKIP sequence counter (TSC) to avoid keystream reuse and packet replay attacks.

4. In addition to the use of ICV in WEP which suffers from message injection and modification attacks, WPA uses MIC. MIC is a strong cryptographic hash function which can only be computed with knowledge of the source and destination MAC addresses, input data stream, MIC key, and the TKIP sequence counter (TSC). This feature of MIC prevents the message from being falsified.

A. TKIP PACKET STRUCTURE

TKIP makes some modifications to the WEP packet structure (Halvorsen & Haugen, 2009). The TKIP packet structure is shown in figure 23.

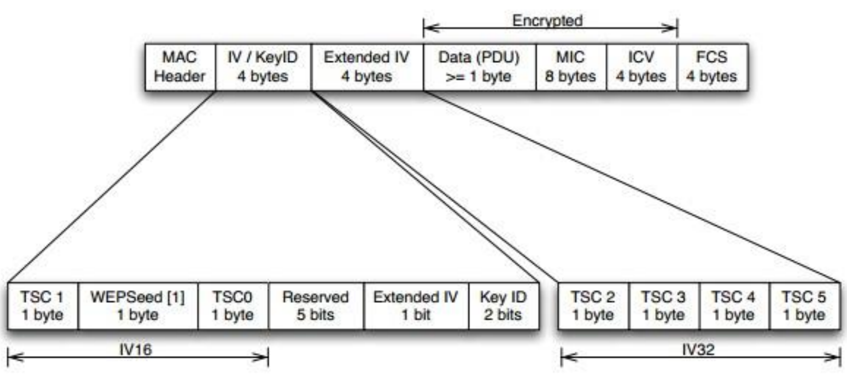


Figure 23: TKIP Packet structure

1. The first part of the TKIP packet structure is the MAC header: The MAC header consists of the sender and receiver MAC addresses.
2. Next it has a 4-byte IV/ KeyID field which differ slightly from WEP.The first 3 bytes serves as the 24-bit WEP IV. It is made up of the 2nd and 1st bytes of the TSC and 1 byte WEP Seed

(inserted to avoid RC4 weak keys). The next 5 bits are reserved for future use. The Extended IV bit is always set to 1 when TKIP is used. The last 2 bits indicated the key ID field (same as in WEP).

3. The Extended IV field consists of 4 bytes which are the remaining four bytes of the TSC.
4. Next follows the Data payload, MIC, and the WEP ICV. These three fields are sent encrypted; all other fields are sent as plaintext.
5. Finally, the FCS is appended to the end of the frame. The FCS is a CRC-32 calculated over the entire frame, including the MAC header.

B. TKIP SEQUENCE COUNTER (TSC)

TSC was designed to address three main weaknesses in WEP IV; they are as follows:

a. The WEP IV was too short (24 bits) and this caused IV reuse.
b. The IV was not used as a sequence counter to prevent message replay.
c. Prepending the IV to the secret key revealed parts of the secret key when weak keys were used.

A 48-bit TSC addresses all these issues (Halvorsen & Haugen, 2009). The larger TSC makes IV reuse not feasible. TSC also functions as a sequence counter, and messages that have equal or lower TSC value than the previous packet is dropped, thus preventing message replay attacks (Halvorsen & Haugen, 2009). Also TSC is constructed to avoid certain class of known weak keys using the 1 byte WEP seed. This prevents keystream attacks.

In addition, TSC increases monotonically (increase by 1) for each packet. Further, TSC is always intialized to 1 when the TKIP temporal key is initialized or refreshed. These features make TSC suitable for a sequence counter. Recall from WEP that there were no requirements for how the IV should be chosen and increased.

C. MESSAGE INTEGRITY CODE (MIC)

One of the biggest problems with WEP was that it suffered from message modification and injection attacks. This was because the ICV which is based on CRC-32 is a linear checksum and it distributes over the entire XOR operation. TKIP uses MIC to defend against message modification and injection attacks. MIC is based on Michael algorithm (Walker, 2005).

Every MIC has three components: a secret authentication key k (shared only between the source and destination nodes), a tagging function, and verification predicate (Walker, 2005).

a. The secret authentication key k is the Pairwise Temporal Key (PTK) which is generated from the Pairwise Master Key (PMK).
b. The tagging function takes the key k and the message M (whose MIC is to be computed) as inputs and generates a tag T which is called the MIC.
c. The sender sends the message M and the generated MIC to the receiver.
d. The receiver computes the PTK (k) of the received message M using its PMK, and generates an MIC for the message.
e. The receiver's computed MIC and the sender's MIC acts as input into the verification predicate.
f. The verification predicate return TRUE if the receiver's MIC is the same as the sender's MIC. Otherwise it returns FALSE which means the message is a forgery.

The advantage of MIC over ICV is that MIC can only be computed with the knowledge of the Key. Also if TKIP detects two failed forgeries in a second, the TKIP algorithm assumes that it is under an active attack. In this case, the station deletes its temporal keys for that message, disassociates, waits for a minute, generates a new PTK for the message, and then re-associates.

While this disrupts communications, it is necessary to thwart active attacks. The bypass for this feature of TKIP is for the attacker to recreate forged messages within intervals of 2 minutes or more.

Recall from figure 23 that the WEP ICV is still calculated on the message. If the WEP ICV check is successful, the MIC is calculated and checked against the received MIC as described above. It is very unlikely for the WEP ICV to compute correct while the MIC check fails. If this happens, it means an active attack is ongoing.

2.3.2 Architecture OF CCMP

CCMP was the second security protocol introduced as a replacement for WEP in the 802.11i amendment. It was adopted by the WPA-2 standard. As opposed to TKIP, CCMP was designed without any consideration for compatibility with old hardware.

CCMP is accomplished through the use of AES block cipher in Counter Mode (CCM) with CBC MAC Mode. Where Counter Mode is used for encryption and CBC is used to generate an MIC.

As CCMP is a totally different design from WEP and TKIP.

Figure 24 shows the CCMP packet, and as can be seen, only the data and MIC are encrypted. The header is very similar to the one used in TKIP, except for some differences. The main difference is the Packet Number (PN). The PN is a 48-bit value used similarly as the TSC of TKIP. The PN is used for replay protection, and to compute a per-packet key.

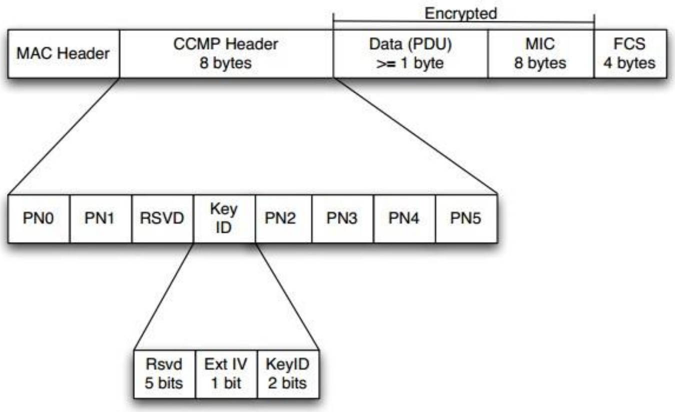


Figure 24: CCMP Packet Structure

Both WPA and WPA-2 provide personal and enterprise editions (Rackley, 2007; Wi-Fi Alliance, 2003; Vivek, 2011). The personal edition uses Pre-Shared Key (PSK) authentication scheme and it is suitable for small office and home wireless network devices (Rackley, 2007). The enterprise edition uses Extensible Authentication Protocol (EAP) scheme by using an external authentication server (RADIUS server) and it is suitable for enterprise wireless networks (Rackley, 2007).

Table 1 below compares WEP, TKIP, and CCMP Key Management and Encryption features:

	WEP	TKIP	CCMP
Encryption Standard	RC4	RC4	AES
Key Size	40 or 104 bits key	128 bits key	128 bits key
Key life determinant	24 bit IV	48 bit IV	48 bit IV
Integrity check	CRC-32	Michael	CCM
Data/Payload header	None	Michael	CCM
Replay Protection	None	Use of IV	Use of IV
Key Installation management	None	EAPOL Based	EAPOL Based

Table 1: Compares WEP, TKIP, and CCMP Key Management and Encryption features

2.3.3 WPA/ WPA-2 PSK EAPOL Handshake

Unlike WEP, WPA/ WPA-2 does not use static keys. Instead it generates dynamic keys on a per packet basis. There are two classes of keys that are generated: The Pairwise Master Key (PMK) and the Group Master Key (GMK). The PMKs are used for unicast or point-to-point communication between two stations while GMKs are used to exchange broadcast or multicast traffic among stations.

A. PAIRWISE MASTER KEY (PMK)

The PMK is a master key. It is not used to encrypt data. Rather, they are used to produce the temporal or transient keys (PTK) which are used for encryption. The concept of master and transient keys are derived from Asymetric or Public Key Crytography as discovered by Diffie and Hellman in their book "New Directions in Crytography" (Menezes et al, 1997).

Once you configure your Access Point or client with WPA or WPA-2 encryption, you input a passphrase which is between 8 to 63 characters long. The PMK is 256 bits long or 64 octets when represented in hexidecimal format (IEEE 802.11, 2012). Most users are familiar with passphrases rather than hexidecimal characters. Hence it is very necessary to have a function that converts the passphrase of between 8 to 63 characters to a 64 octet hexidecimal character. This passphrase to PMK mapping is achieved using the Password Based Key Derivation Function (PBKDF2) (Akin, 2005; IEEE 802.11, 2012).

PBKDF2 is based on RFC 2898 and it is defined as

PMK = PBKDF2 (PassPhrase, ssid, ssidLength, 4096, 256).

PBKDF2 takes the passphrase entered by the user, the ssid and ssidLength of the Access Point. It then hashes these 4096 times to output a 256 bit Pre-Shared key called the PMK (IEEE 802.11, 2012). This generated PMK is installed on the client. Similar, the AP goes ahead to take the same Passphrase (Password) entered by the user on the AP to generate the same PMK and install it on the AP (Vivek, 2011) as shown in figure 25.

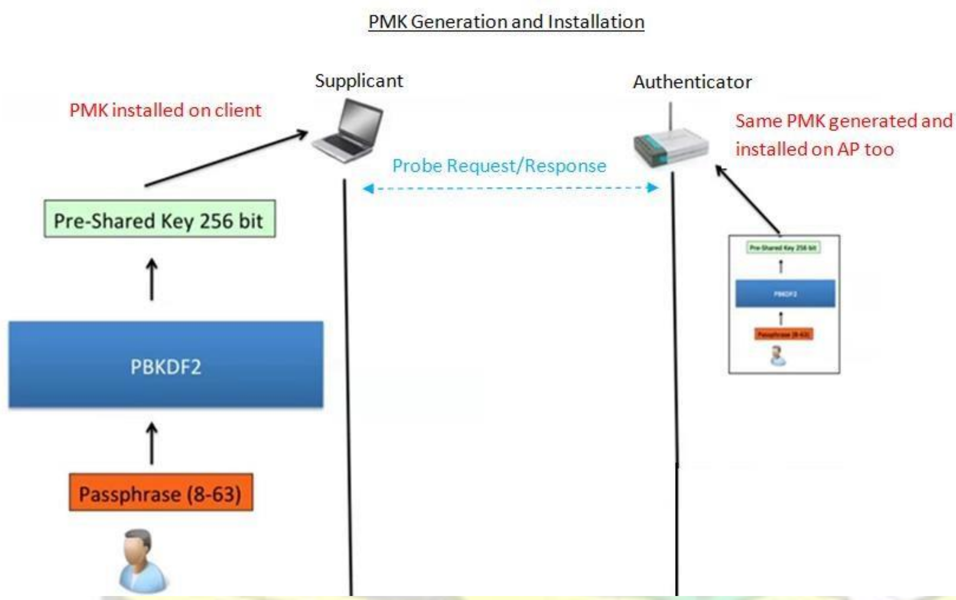


Figure 25: PMK Generation and Installation mechanism

B. Group Master Key (GMK)

The GMKs are group master keys. They are not used to encrypt data. Rather, they are used to produce the Group temporal or transient keys (GTKs) which are used for encrypting multicast and broadcast packets.

C. Pairwise Transient Key (PTK)

In order to obtain the PTKs or GTKs, a four-way or two-way EAPOL Handshake respectively are performed between the Access Point (Authenticator) and client (Supplicant) after the PMKs or GMKs have been installed (Akin, 2005; IEEE 802.11, 2012, Sithirasenan et al, 2005).

2.3.4 Four-way EAPOL Handshake to generate and install PTK

1. As soon as the PMKs are installed, the authenticator generates a long random value called Authenticator Nounce (ANounce) as shown in figure 26. The ANounce is sent as part of Message 1 to the client in figure 27.

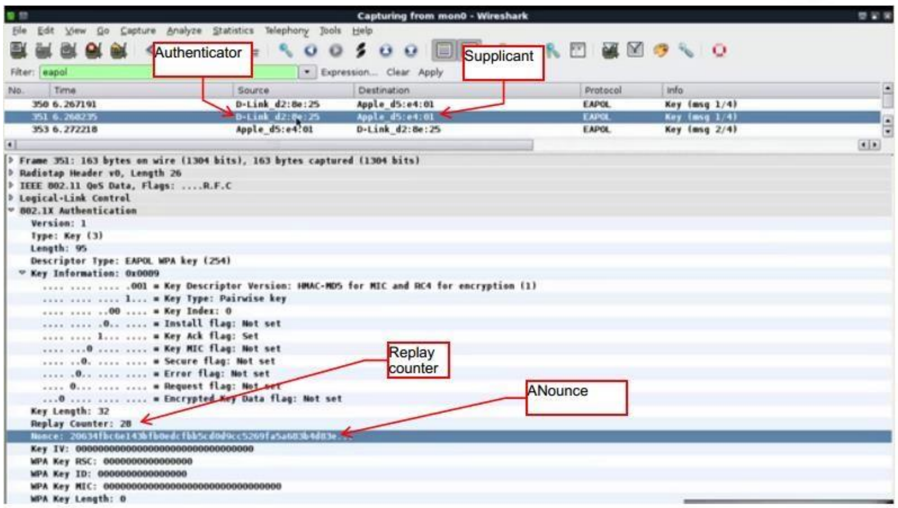


Figure 26: A random ANounce sent by the authenticator to the supplicant

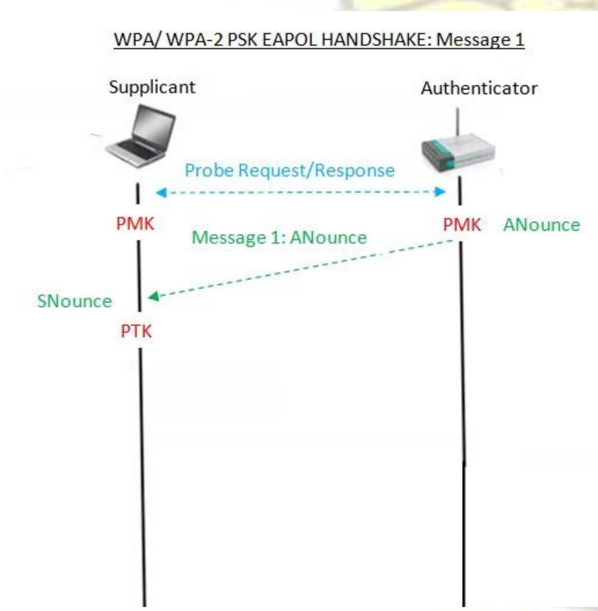


Figure 27: Message 1 of the EAPOL Handshake sent from the Authenticator to the Supplicant

2. As soon as the client receives Message 1 from the AP, it goes ahead to generate its own long random value or message called Supplicant Nounce (SNounce) as shown in figure 27.
3. The client with the knowledge of the ANounce, SNounce, client MAC Address, and AP MAC Address; goes ahead to calculate its PTK as follows:

   PTK = Function (PMK, ANounce, SNounce, Authenticator Mac Address, Supplicant Mac Address).

4. The generated PTK is 512 bits long. The first 256 bits is used to protect the EAPOL Handshake while the remaining 256 bits is used to protect the actual Data transfer between the client and AP (Akin, 2005) as shown in figure 28.

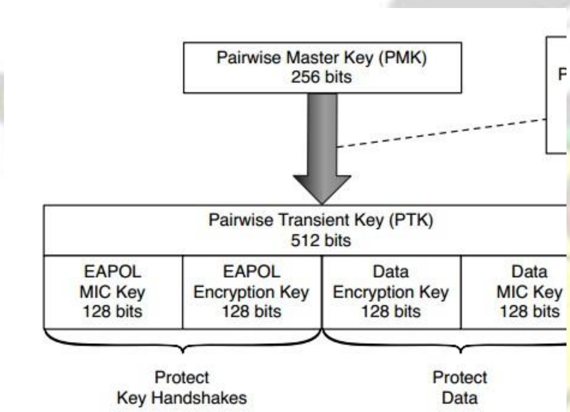


Figure 28: The functional parts of the 512 bits generated PTK

5. The client then computes a 128 bits MIC called the EAPOL MIC Key over the entire PTK and over the entire EAPOL frame to be sent to the authenticator. The supplicant then sends the SNounce plus the computed MIC to the authenticator in Message 2 of the EAPOL Handshake as shown in figures 29 and 30.

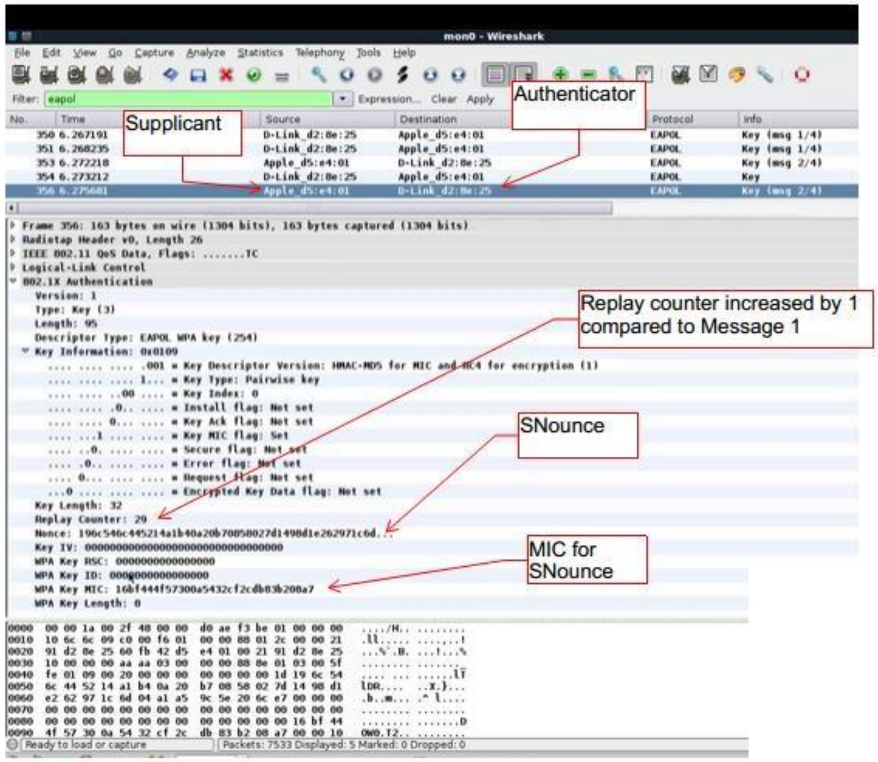


Figure 29: A random ANounce sent by the supplicant to the authenticator

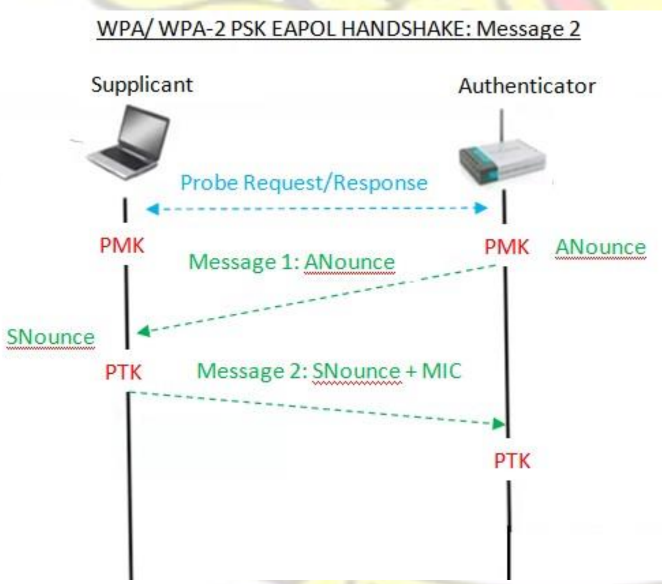


Figure 30: Message 2 of the EAPOL Handshake sent from the Supplicant to the Authenticator

6. The authenticator upon receiving Message 2 (SNounce and MIC) goes ahead to compute its own PTK using its PMK, the SNounce, ANounce, client MAC Address, and AP MAC Address.

PTK = Function (PMK, ANounce, SNounce, Authenticator Mac Address, Supplicant Mac Address).

7. After computing the PTK, the Authenticator goes ahead to compute a 128 bits MIC for the PTK it derived and over the EAPOL frame it received in Message 2 from the supplicant.
8. If there is a match with the MIC sent by the client, the authenticator knows that the supplicant also ended up deriving the same PTK and hence supplicant has the same PMK as the authenticator.
9. Next the authenticator sends Message 3 which is the Key installation message as shown in figures 31 and 32 to the supplicant after the success of step 8. Otherwise, it sends a deauthentication message to the client if the MICs did not match. In addition, the authenticator appends an MIC to Message 3 for the supplicant to mutually authenticate the AP too. Message 3 tells the supplicant to go ahead to install and use its derived PTK for any future transactions until the connection breaks or a new PTK is derived.

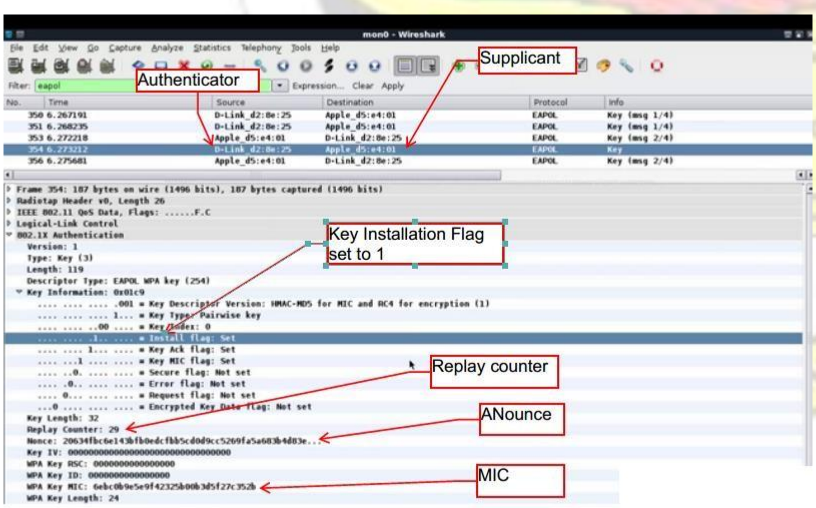


Figure 31: Message 3 of the EAPOL Handshake sent from the Authenticator to the Supplicant as captured with Wireshark

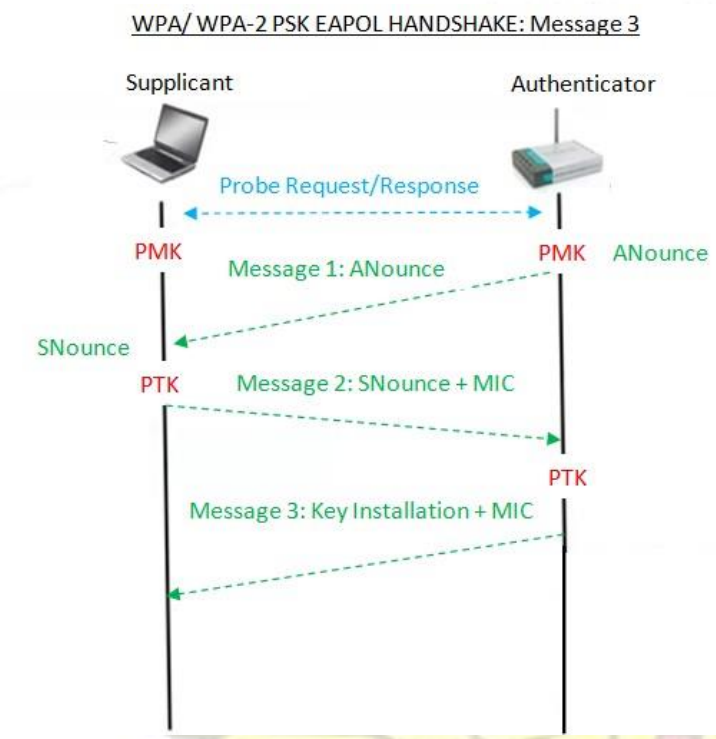


Figure 32: Message 3 of the EAPOL Handshake sent from the Authenticator to the Supplicant

10. The Supplicant upon receiving Message 3 can go ahead to first verify the authenticity of the message by using the MIC sent by the authenticator after which it goes ahead to install its derived PTK and then send Message 4 which is the Key installation Acknowledgement message to the Authenticator as shown in figure 33.

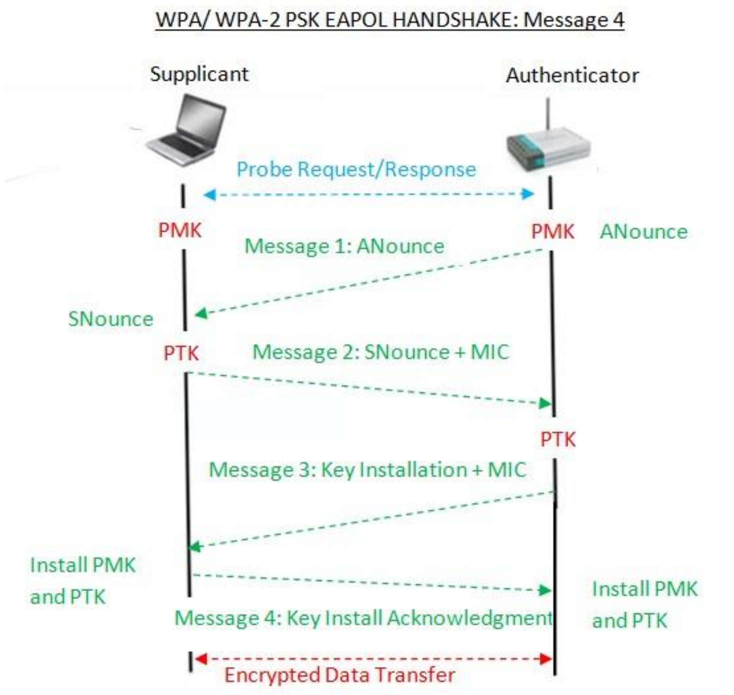


Figure 33: Message 4 of the EAPOL Handshake sent from the Supplicant to the Authenticator

11. After the successful installation of the Pairwise Master Keys (PMKs) and Pairwise Transient Keys (PTKs) by both the Authenticator and Supplicant, encrypted data transfer using the PTK now starts to take place between the Access Point and the Client as shown in figure 33.

2.4 IEEE 802.11 WI-FI PROTECTED ACCESS ENTERPRISE AUTHENTICATION PROTOCOL (WPA/ WPA2-EAP)

Architecture

WPA/ WPA2 supports enterprise edition using an Extensible Authentication Protocol (EAP) (Madjid & Mahsa, 2005). As the name suggest, EAP outsources the authentication scheme to an external server instead of the Access Point (Rackley, 2007). The external server or authentication server runs a standard Authentication, Authorization, and Accounting (AAA) protocol such as Remote Access Dial-In User Service (RADIUS) or DIAMETER (DIAMETER has twice the functions of RADIUS) (Madjid & Mahsa, 2005).

The Access Point acts as an Authenticator; it is responsible for forwarding client information in the form of request to the Authentication server, waits for the response from the server and passes it on to the client (supplicant) (Madjid & Mahsa, 2005).

EAP messages operate at the data link layer of the OSI Model and are carried over LAN between the Supplicant and Authenticator. Between the Authenticator and the Authentication server, EAP messages are carried over AAA protocols such as RADIUS.

2.4.1 EAP over LAN Message Types

There are four types of messages in EAP between the supplicant and the authenticator. They include the following:

- EAP Request Message: This message is sent by the authenticator to the supplicant for the supplicant to prove its identity.
- EAP Response Message: The supplicant proves its identity to the authenticator with an EAP Response message.
- EAP Success Message: This message is sent by the authenticator to the supplicant after the authentication server has accepted the identity of the supplicant.

- EAP Failure Message: This message is sent by the authenticator to the supplicant after the authentication server has rejected the identity of the supplicant.

2.4.2 EAP over RADIUS Message Types

There are four types of messages in EAP between the authenticator and authentication server. They include the following:

- RADIUS Request Message: This message is sent by the authenticator to the authentication server to forward the request from the supplicant to the authentication server.
- RADIUS Access Challenge Message: This message is sent by the authentication server to the authenticator. It is generally used to prove the identity of the authenticator or supplicant or to perform some sort of negotiation between the authenticator and supplicant to the authentication server.
- RADIUS Access Accept Message: This message is sent by the authentication server to the authenticator to indicate a successful grant of access to the authenticator's or supplicant's identity request.
- RADIUS Access Reject Message: This message is sent by the authentication server to the authenticator to indicate a rejection to the authenticator's or supplicant's identity request.

Figure 34 illustrates the exchange of EAP messaging types between the supplicant, authenticator, and authentication server:

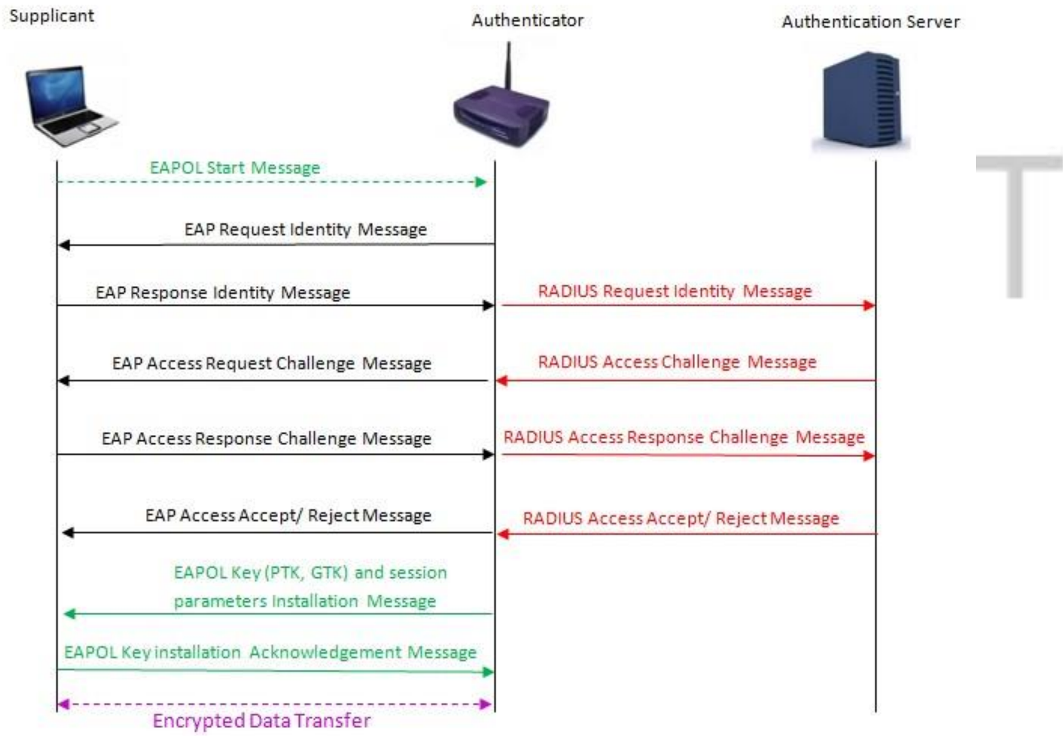


Figure 34: The exchange of EAP messaging types between the supplicant, authenticator, and authentication server

2.4.3 EAP Authentication Methods

EAP itself does not perform the actual authentication (Madjid & Mahsa, 2005). Rather, it is augmented with authentication methods that have their own requirements and procedures. These EAP methods are carried within the "type-field" of the EAP Request and Response messages (Madjid & Mahsa, 2005). Table 2 provides a latest list of EAP Authentication methods and their type-field values found at IANA website for EAP numbers (EAPIANA) (http://www.iana.org/assignments/eap-numbers/eap-numbers.xhtml#eap-numbers-4).

Value	Description	Reference
0	Reserved	
1	Identity	[RFC3748]
2	Notification	[RFC3748]
3	Legacy Nak	[RFC3748]
4	MD5-Challenge	[RFC3748]
5	One-Time Password (OTP)	[RFC3748]
6	Generic Token Card (GTC)	[RFC3748]
7	Allocated	[RFC3748]
8	Allocated	[RFC3748]
9	RSA Public Key Authentication	[William_Whelan]
10	DSS Unilateral	[William_Nace]
11	KEA	[William_Nace]
12	KEA-VALIDATE	[William_Nace]
13	EAP-TLS	[[Aboba]]
14	Defender Token (AXENT)	[Michael_Rosselli]
15	RSA Security SecurID EAP	[Magnus_Nystrom]
16	Arcot Systems EAP	[Rob_Jerdonek]
17	EAP-Cisco Wireless	[Stuart_Norman]
18	GSM Subscriber Identity Modules (EAP-SIM)	[RFC4186]
19	SRP-SHA1	[James_Carlson]
20	Unassigned	
21	EAP-TTLS	[RFC5281]
22	Remote Access Service	[Steven_Fields]
23	EAP-AKA Authentication	[RFC4187]
24	EAP-3Com Wireless	[Albert_Young]
25	PEAP	[Ashwin_Palekar]
26	MS-EAP-Authentication	[Ashwin_Palekar]
27	Mutual Authentication w/Key Exchange (MAKE)	[Romain_Berrendonner]
28	CRYPTOCard	[Stephen_M_Webb]
29	EAP-MSCHAP-V2	[Darran_Potter]
30	DynamID	[Pascal_Merlin]
31	Rob EAP	[Sana_Ullah]
32	Protected One-Time Password	[RFC4793] [Magnus_Nystrom]
33	MS-Authentication-TLV	[Ashwin_Palekar]
34	SentriNET	[Joe_Kelleher]
35	EAP-Actiontec Wireless	[Victor_Chang]
36	Cogent Systems Biometrics Authentication EAP	[John_Xiong]
37	AirFortress EAP	[Richard_Hibbard]
38	EAP-HTTP Digest	[Oliver_K_Tavakoli]
39	SecureSuite EAP	[Matt_Clements]
40	DeviceConnect EAP	[David_Pitard]

Table 2: Provides a latest list of EAP Authentication methods and their type-field values

Value	Description	Reference

41	EAP-SPEKE	[Don_Zick]
42	EAP-MOBAC	[Tom_Rixom]
43	EAP-FAST	[RFC4851]
44	ZoneLabs EAP (ZLXEAP)	[Darrin_Bogue]
45	EAP-Link	[Don_Zick]
46	EAP-PAX	[T_Charles_Clancy]
47	EAP-PSK	[RFC4764]
48	EAP-SAKE	[RFC4763]
49	EAP-IKEv2	[RFC5106]
50	EAP-AKA'	[RFC5448]
51	EAP-GPSK	[RFC5433]
52	EAP-pwd	[RFC5931]
53	EAP-EKE Version 1	[RFC6124]
54	EAP Method Type for PT-EAP	[RFC7171]
55	TEAP	[RFC7170]
56-191	Unassigned	
192-253	Unassigned	
254	Reserved for the Expanded Type	[RFC3748]
255	Experimental	[RFC3748]
256-4294967295	Unassigned	

Table 2 continuation provides a latest list of EAP Authentication methods and their type-field values

When the supplicant or authentication server receives an EAP Request or Response message, depending on the type-field, it will forward it to its corresponding upper layer model for processing as shown in figure 35. Otherwise it will return an NAK (type-field = 3) message to the peer if it does not support the type of EAP method been supplied by the peer.

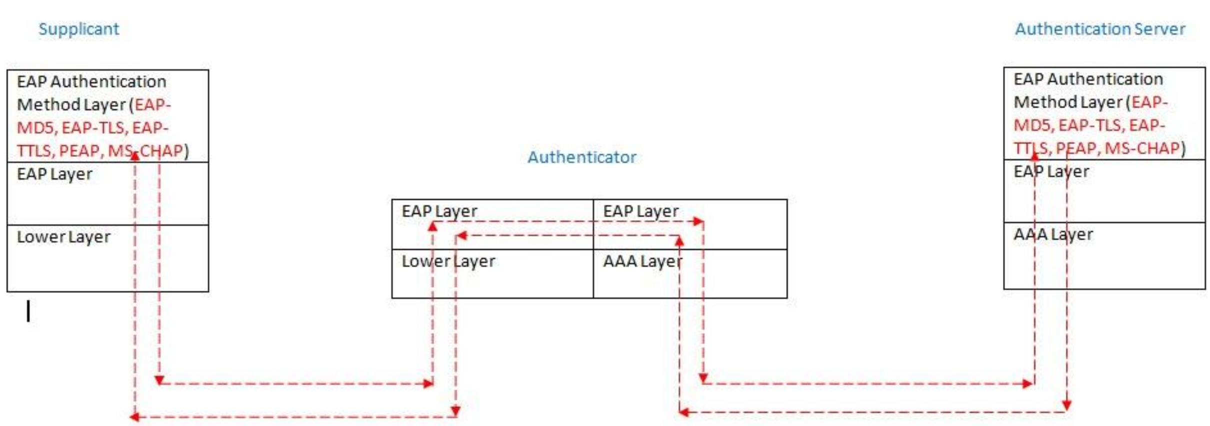


Figure 35: The EAP Layering model for carrying EAP Authentication Method messages

The most widely used EAP types are EAP-MD5, PEAP, EAP-TLS, EAP-TTLS, LEAP, and EAPFAST (Vivek, 2011; Madjid & Mahsa, 2005). The proceeding sections, describes the architecture of some of these EAP types.

2.4.4 EAP-MD5 Architecture and working mechanism

1. The supplicant is configured to support EAP-MD5 as shown in figure 36 below:

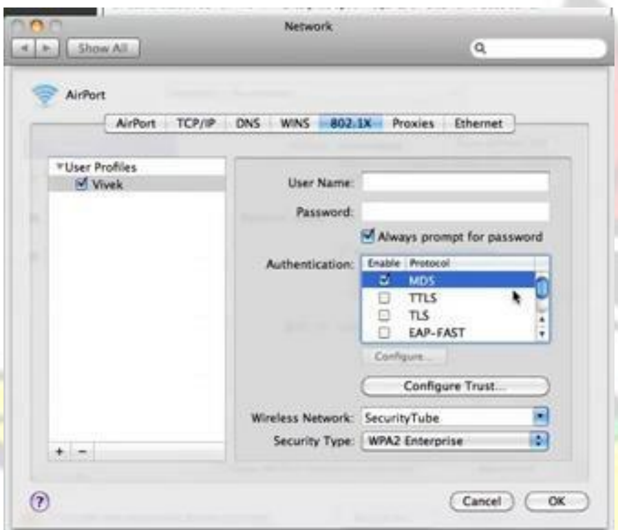


Figure 36: An interface of apple laptop been configured to support EAP-MD5

2. The EAP Authentication method in the RADIUS server is also set to EAP-MD5.

3. The supplicant inputs a username and a password (passphrase) as shown in figure 37.

Figure 37: The user inputting username/ password in the supplicant in order to authenticate with the RADIUS server

4. The same username and passphrase is set on the RADIUS server by using the command

"*vim users"* (see Appendix B on how to setup a RADIUS server in BACKTRACK5).

For example, the username "Securitytube" and password "demo12345" as shown in figure 38.


```
#
# This is a complete entry for "steve". Note that there is no Fall-Through
# entry so that no DEFAULT entry will be used, and the user will NOT
# get any attributes in addition to the ones listed here.

SecurityTube    Cleartext-Password := "demo12345"

#
#steve  Cleartext-Password := "testing"
#       Service-Type = Framed-User,
#       Framed-Protocol = PPP,
#       Framed-IP-Address = 172.16.3.33,
#       Framed-IP-Netmask = 255.255.255.0,
#       Framed-Routing = Broadcast-Listen,
#       Framed-Filter-Id = "std.ppp",
#       Framed-MTU = 1500,
#       Framed-Compression = Van-Jacobsen-TCP-IP

#
# This is an entry for a user with a space in their name.
# Note the double quotes surrounding the name.
#
#"John Doe"     Cleartext-Password := "hello"
#               Reply-Message = "Hello, %{User-Name}"
```

Figure 38: The same supplicant credentials input into the user database of the RADIUS server

5. The shared secret between the RADIUS server and the Access Point is set to be the same as shown in steps 2 and 5 in Appendix B.

6. As soon as the RADIUS server is started as shown in step 6 in Appendix B, the supplicant sends and EAP Start message to the AP.

7. The Authenticator sends an EAP Identity Request packet to the supplicant as shown in figure 39.

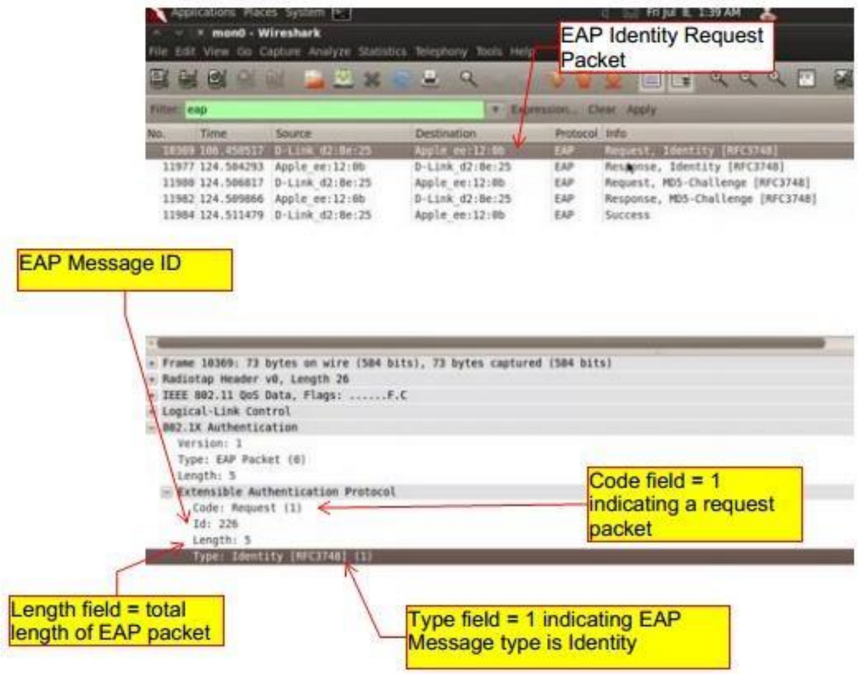


Figure 39: An EAP Identity Request Packet sent from Authenticator to Supplicant

8. The supplicant upon receiving the request issues a response to the EAP Identity request as shown in figure 40.

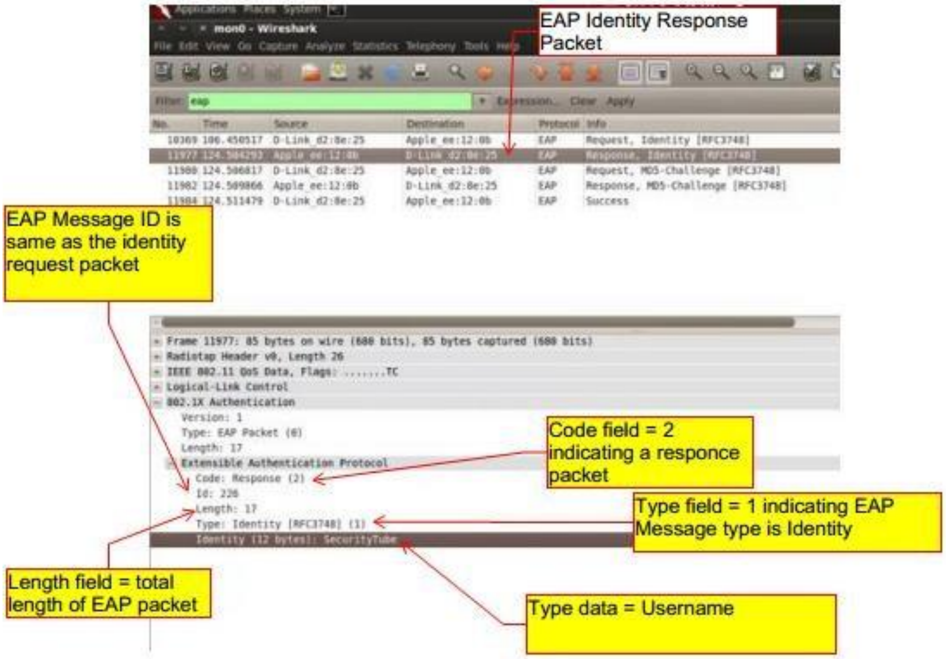


Figure 40: An EAP Identity Response Packet sent from Supplicant to Authenticator

9. Upon receipt of the EAP Identity Response Packet from the supplicant, the Authenticator creates an EAP Access-Request packet and sends it on UDP port 1812 to the RADIUS Server as shown in figure 41. The packet contains two very important fields:

- EAP Message Attribute: has code field = 79 and contains the Identity Response packet received from the supplicant.
- EAP Message Authenticator: has code field = 80 and is calculated as an HMACMD5 of the whole packet using the shared secret between the authenticator and RADIUS server.

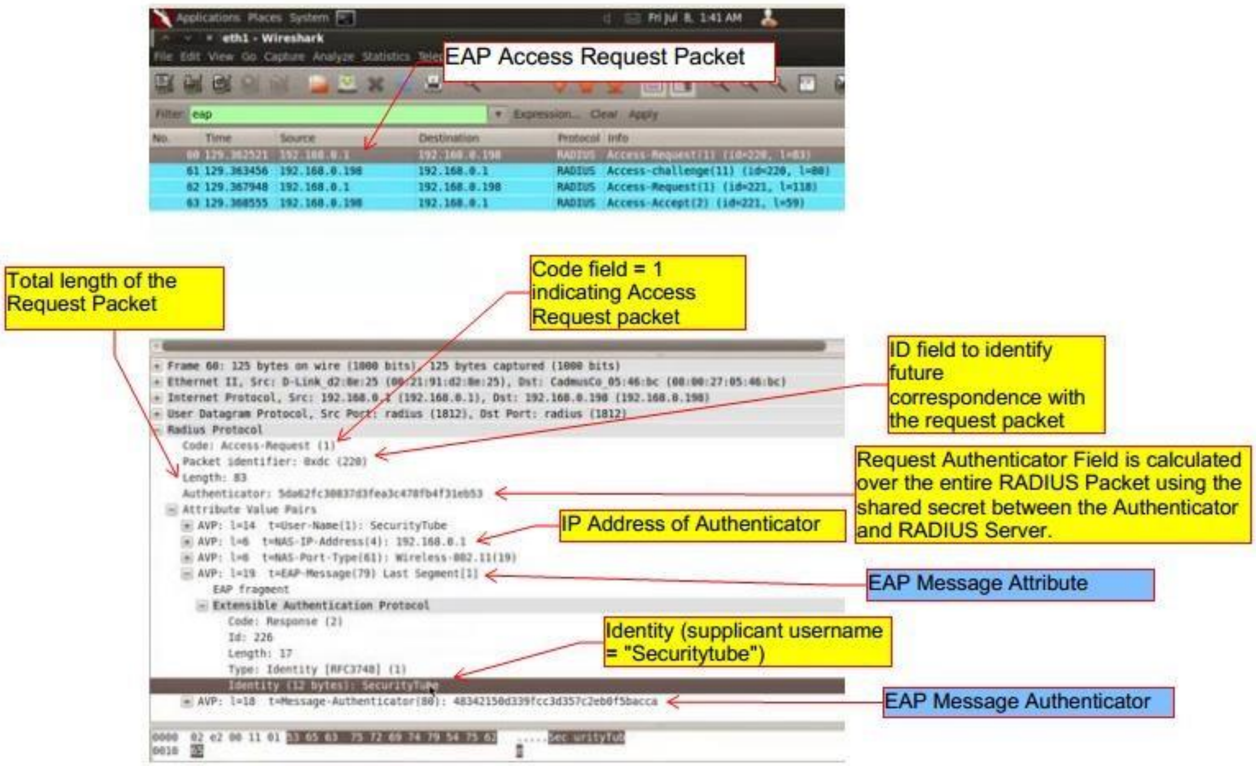


Figure 41: RADIUS EAP Access Request packet sent from Authenticator to RADIUS server

10. The RADIUS server upon receiving the Access Request packet from the Authenticator goes ahead to check the Request Authenticator Field and the Message Authenticator Field in order to prove the Authorization of the Access Point. If the checks return true, the RADIUS server generates a long random Challenge message called the "RADIUS Challenge" and sends towards the supplicant as shown in figure 42.

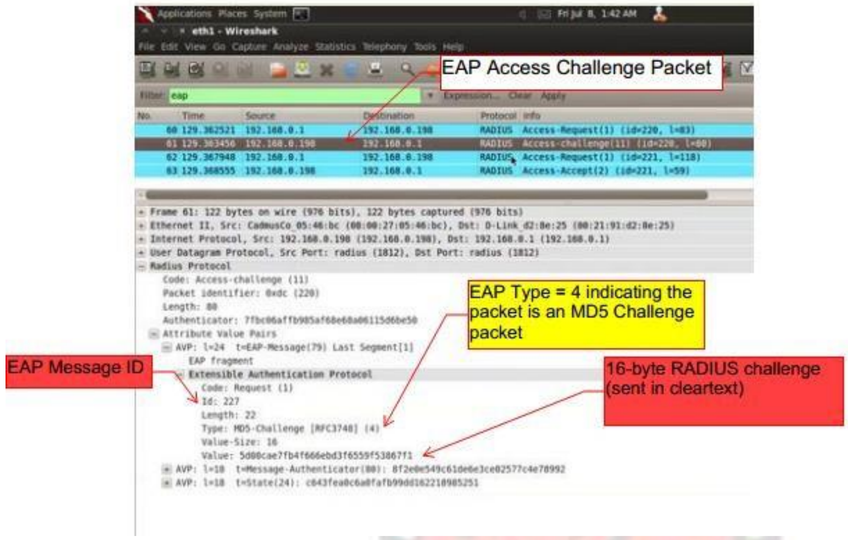


Figure 42: The RADIUS Access Challenge Message sent in cleartext from the Server to the Authenticator

11. The authenticator upon receiving the RADIUS Challenge, checks the Request Authenticator Field and the Message Authenticator Field in order to prove the authenticity and integrity of the Authentication Server. If the checks return true, it extracts the EAP message value and crafts an EAP Challenge packet of type MD5 Challenge and relays the same 16-byte challenge value to the supplicant as shown in figure 43.

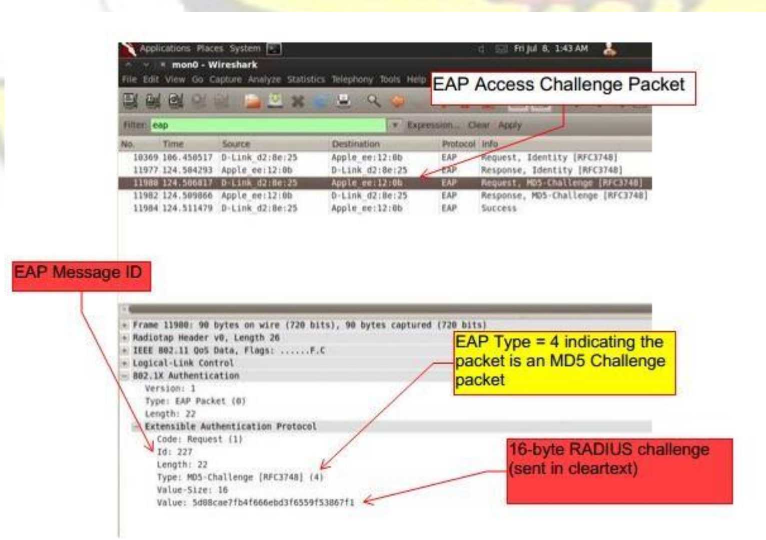


Figure 43: The RADIUS Access Challenge message encapsulated in an EAPOL packet and sent from the Authenticator to the Supplicant

12. The Supplicant upon receiving the message extracts the value of the Challenge and the ID of the EAP message. It then calculates the response value of this challenge by performing an MD5 hash as follows:

EAP MD5 Hash value = (EAP-Message ID + User Password + RADIUS Challenge value).

This MD5 Hash value is inserted into an EAP Message of type 4 with same EAP-Message ID and code set to '2' to indicate it is a response EAP packet of type MD5 Challenge. The packet is again transmitted back to the Authenticator as an EAP Access Response Message as shown in figure 44.

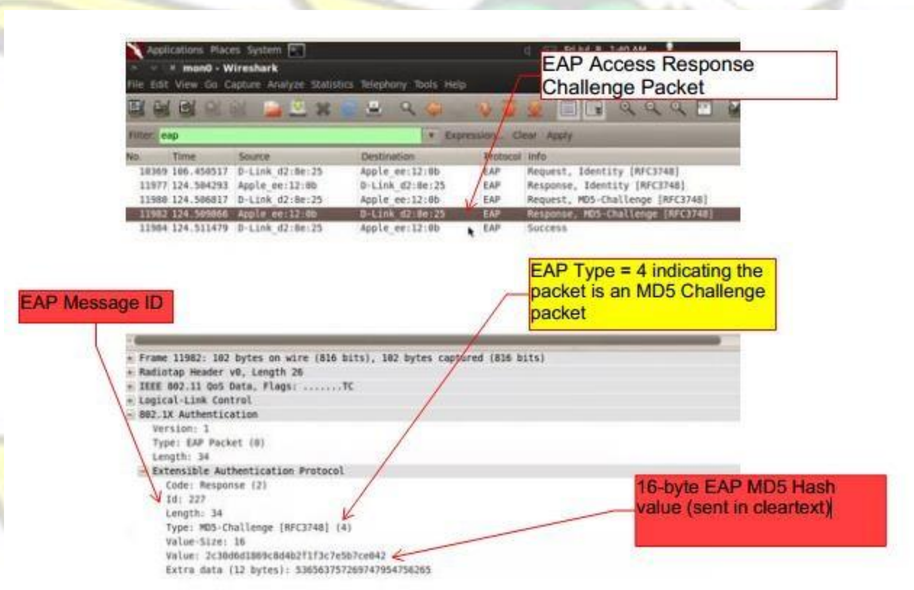


Figure 44: The EAP Access Response Challenge message sent from supplicant to the authenticator

13. The Authenticator forwards this EAP Access Response packet by encapsulating it in an Access-Request Packet to the RADIUS Server as shown in figure 45.

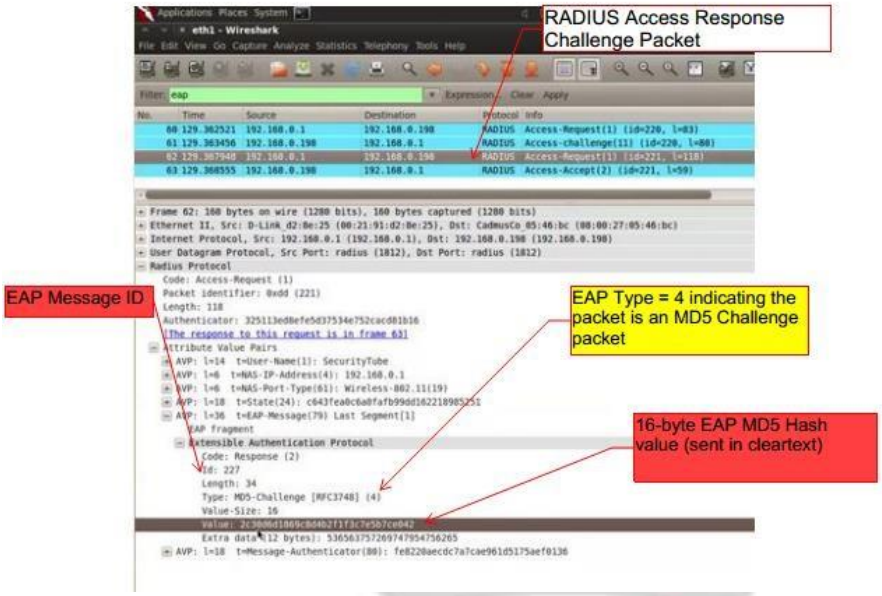


Figure 45: The RADIUS Access Response Challenge message sent from authenticator to RADIUS Server

14. The RADIUS server verifies the Request Authenticator Field and the Message Authenticator Field to prove the authenticity and integrity of the packet. It then extracts the Challenge/Response value from the EAP-Message attribute. The response to the challenge is identified by matching the EAP Message ID which was issued at the time of generating the Access-Challenge packet. The server now performs a similar MD5 hash as follows:

EAP MD5 Hash value = (EAP-Message ID + User Password (from the user database) + RADIUS Challenge value).

15. If the MD5 Hash value computed by the server matches the MD5 Hash value sent by the supplicant, an Access-Accept message with other configuration parameters with an EAP Message attribute of type EAP-Success is created and sent to the Authenticator as shown in figure 46.

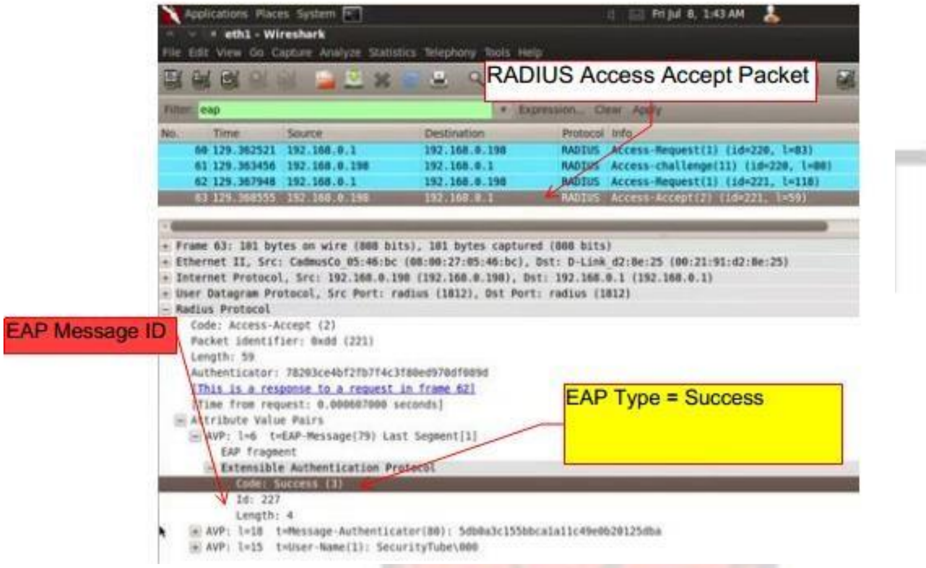


Figure 46: The RADIUS Access Accept message sent from the RADIUS server to the Authenticator

16. The authenticator in turn performs the usual checks on the Request Authenticator and the Message Authenticator attributes and on success forwards the EAP-Success message to the supplicant as shown in figure 47.

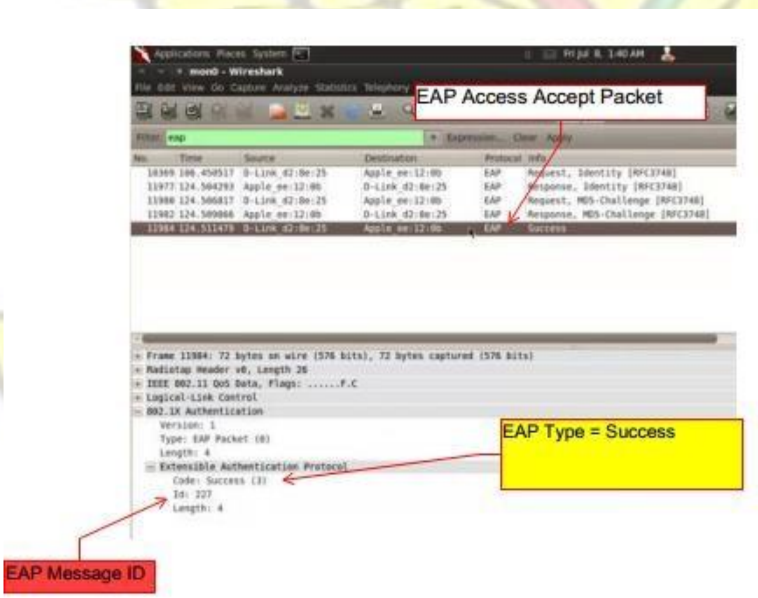


Figure 47: The EAP Access Accept message sent from the Authenticator to the Supplicant

17. At this point, the supplicant is granted access to the remote network or further negotiations on Network Layer might take place. Other negotiations such as Accounting might also take place.
18. If both the MD5 values had not matched, an Access-Reject message with an EAP Message attribute of type EAP-Failure would have been created by the RADIUS server and sent to the Authenticator.
19. The authenticator would have in turn performed the usual checks on the Request Authenticator and Message Authenticator attributes and on success would forward the EAP-Failure message to the supplicant.
20. The supplicant at this point would have been denied access to the network and a suitable message describing the error would be displayed to the user per the error code returned by the authenticator.

2.4.5 Other EAP Inner Authentication Architectures and Mechanisms

Besides EAP-MD5, there are other EAP Inner Authentication mechanisms. These include Microsoft Challenge-Handshake Authentication Protocol (MSCHAPv2). All these protocols are challenge and response messages which may be susceptible to offline dictionary attacks. To improve upon the security, the IEEE 802.11 group recommended that these Inner Authentication protocols are carried over secured Tunnels called Transport Layer Sections (TLS). These tunnels are setup between the Authentication server and the supplicant (Aboba & Simon, 1999) as shown in figure 48.

Inner Authentication Mechanisms (MS-CHAP, MD5-CHAP, EAP-MD5, PAP)
Carried over Secured Tunnels (TLS)
Encapsulated as EAP messages (EAP-TLS)
Transmitted as EAPOL, RADIUS, or DIAMETER messages over the LAN

Figure 48: Deploying inner authentication mechanisms (such as CHAP,PAP) over secured TLS channels in EAP framework

These tunnels are setup using the concept of Public Key Cryptography (PK1) (Stallings, 2006). The idea is to ensure confidentiality and integrity of data between the Authentication server and Supplicant.

PKI uses the idea of public and private keys. Either the public or private key is used for encryption while the other is used for decryption. Think of public key as a composite number while the private key as two prime numbers multiplied to get the composite number. While it is very easy for computers to multiply two big prime numbers to obtain a composite number, it is virtually impossible (probably may take a thousand years) for computers to derive or bruteforce the two big prime numbers given their composite number. The composite number or public key is generated by the originator and distributed to all participants who wish to take part in the communication. The Private Key is kept private with the originator and are not shared. Participants will encrypt data with the public key and send back to the originator who has the private key. Although the encryption/ decryption algorithm is known by all participants including a hacker, only the originator who has the private key can decrypt the data. Thus confidentiality has been ensured.

To ensure integrity of data, PKI uses digital certificates. A digital certificate is where an originator of a message encrypts the message with its private key and only stations in possession of the originator's public key can decrypt the data (Stallings, 2006). There are two types of digital certificates: server-side certificates and client-side certificates. Server-side certificates are originated by the server while client-side certificates are originated by the client.

A. SERVER-SIDE DIGITAL CERTIFICATE

Server-side certificates are originated by the server. For example, the authentication server uses its private key to encrypt the inner authentication protocol such as MSCHAPv2 and then re-encrypts it with the public key of the legitimate supplicant. The earlier ensures integrity whilst the latter ensures confidentiality of the MSCHAPv2 message over the air interface. Only the legitimate supplicant who possesses the private key can decrypt the message and then use the public key of the authentication server to further decrypt it. The supplicant then responds to the challenge of the inner authentication protocol, encrypts it with the public key of the legitimate server and transmits its back to the server.

A well-known vulnerability in implementing server-side certificates is that because authentication servers serve many supplicants at a time, it is computationally expensive for the server to reencrypt each packet with the public key of every supplicant (Vivek, 2011). Hence implementers usually ignore the confidentiality aspect of the message (Stallings, 2006). This flaw will be further exploited in this thesis work in an attempt to hack into WLAN as any station in possession of the public key of the server may be able to decrypt the server-side digital certificates.

B. CLIENT-SIDE DIGITAL CERTIFICATE

Client-side certificates are originated by the client to validate or authenticate the server. Here, the supplicant uses its private key to encrypt the responds message of the inner authentication protocol such as MSCHAPv2 and then re-encrypts it with the public key of the legitimate server. Only the legitimate server in possession of the private key can decrypt the message and then use the public key of the legitimate supplicant to further decrypt the message. The authentication server will then respond to the legitimate supplicant by encrypting it with the public key of the supplicant. If the supplicant decrypts the message and finds the message to be same as what it expected, it has successfully validated the server with its client-side certificate.

To provide mutual authentication, EAP Methods must provide mandatory support for both serverside and client-side certificates (Stallings, 2006; Aboba & Simon, 1999). As at the time of writing this thesis, only EAP-TLS (EAP method id = 13) provide mandatory support for both server-side and client-side certificates. EAP Tunneled Transport Layer Security (EAP-TTLS) and Protected

EAP (PEAPv0) provide mandatory support for server-side certificates while the use of client-side certificates is optional. PEAPv0 comes natively installed on all windows Operating Systems and hence has the highest WLAN deployment worldwide than any other EAP Method (Vivek, 2011).

CHAPTER 3 EXPERIMENTATION AND THESIS SURVEY

3.1 Overview

This chapter is the experiments. It describes the step by step experimental procedure that was used in an attempt to discover the vulnerabilities in the security protocols of WLANs. It also provides data of security protocols configured on 1,271 Access points surveyed in Ghana.

3.2 Laboratory setup requirements

The laboratory was setup with two laptops (one as the attacker's laptop and the second as the victim's laptop), an access point, an authentication server, and a sniffing wireless card as shown in figure 49. The detailed procedure to setup the laboratory is described in Appendix B.

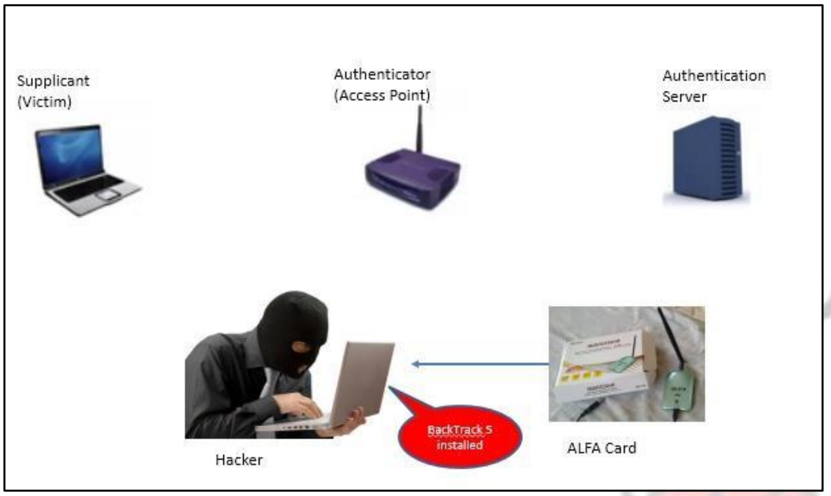


Figure 49: The laboratory setup diagram

3.3 Vulnerabilities in IEEE 802.11 WEP Security Protocol

In this section, three security vulnerabilities were discovered in WEP through experiments.

A. WEP AUTHENTICATION IS VERY EASY TO COMPROMISE

It was proved that it is possible to authenticate to a WEP network without knowledge of the Password. The below experiment provides the evidence of this vulnerability:

1. BackTrack 5 was used to eavesdrop to discover the various WLANs that are broadcasting their SSIDs and including those whose SSIDs had been turned off. Figure 50 shows the use of the command “airodump-ng mon0” in BackTrack5 to eavesdrop on a WEP secured WLAN whose SSID is TP-LINK_POCKET_3020_ 7654BF.

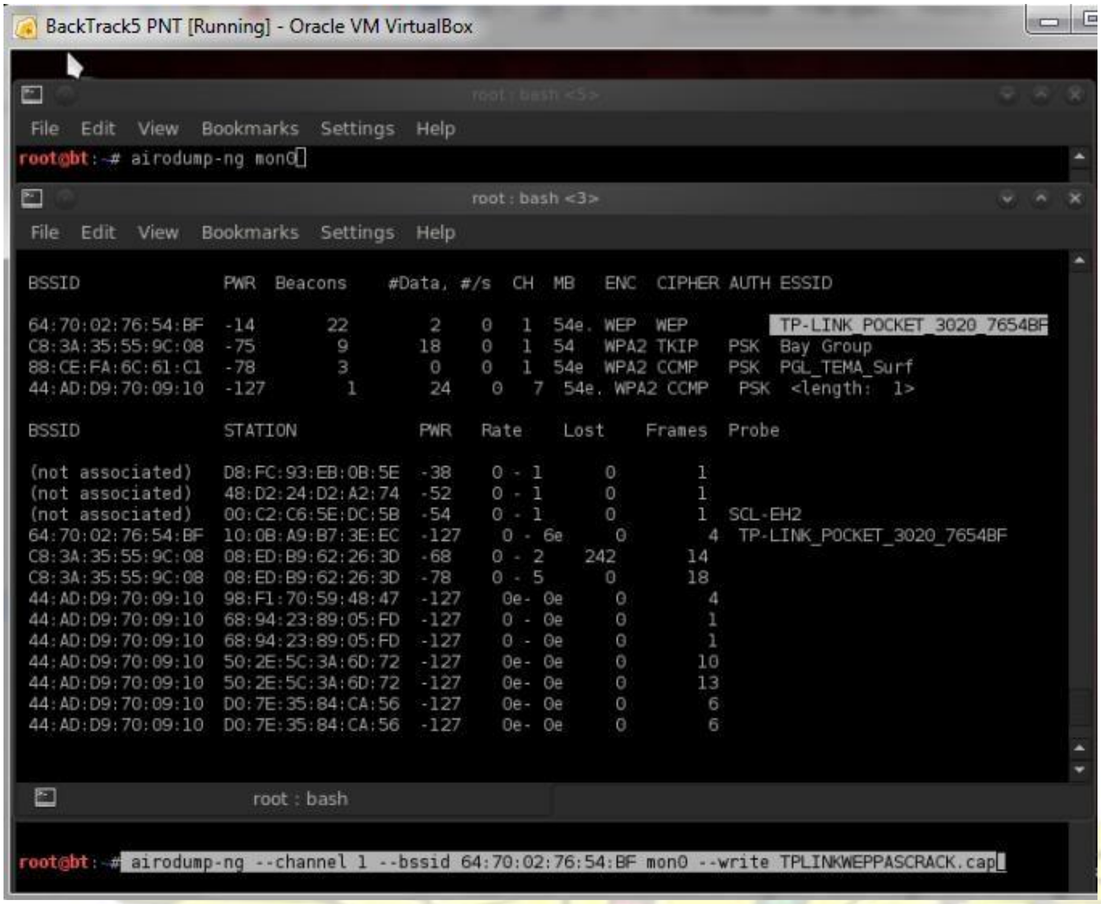
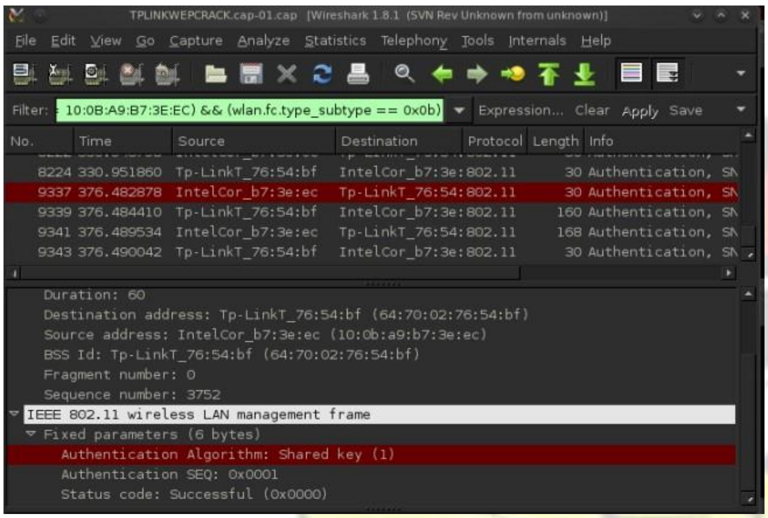


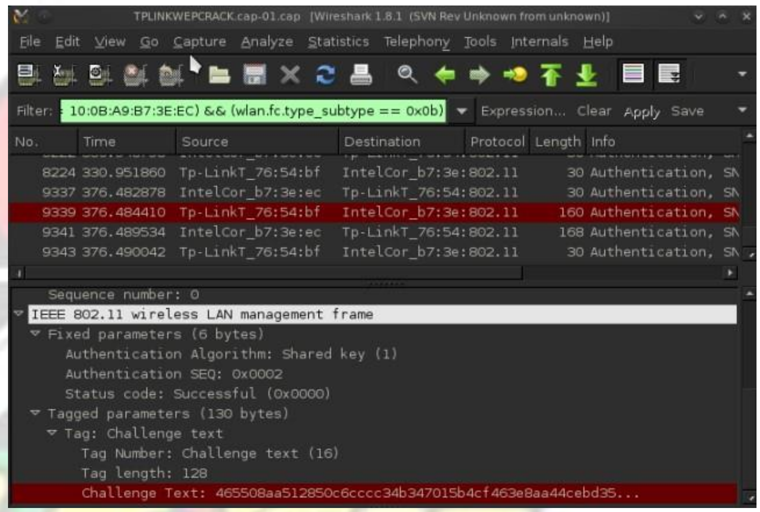
Figure 50: "airodump-ng" command used to monitor available WLAN networks

2. During authentication phase, the packets that was exchanged between a legitimate client and a legitimate access point was eavesdropped and written to a .cap file with the “--write” command in BackTrack5. Figure 51 the four captured authentication packets.

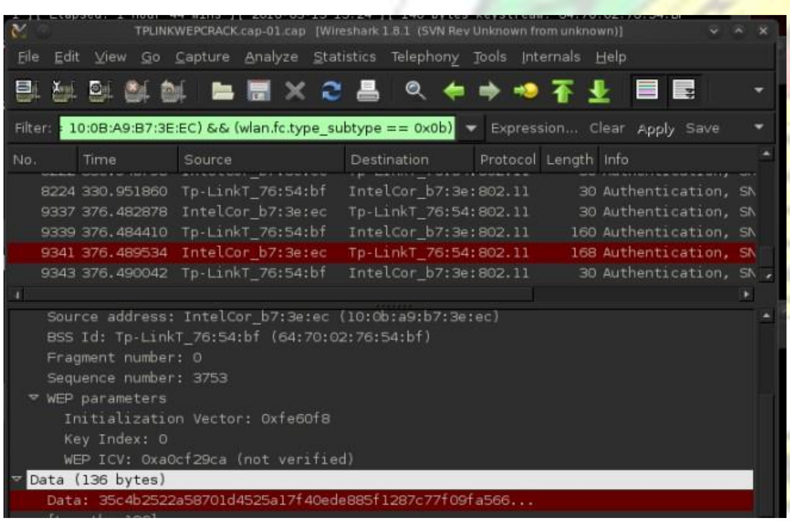
Message 1:**Authentication Request Packet** from legitimate client to legitimate AP



Message 2: **Challenge Plaintext Packet** from AP to legitimate client



Message 3: **Challenge Response Ciphertext Packet** from legitimate client to AP



Message 4: **Authentication Success Packet** from AP to legitimate client

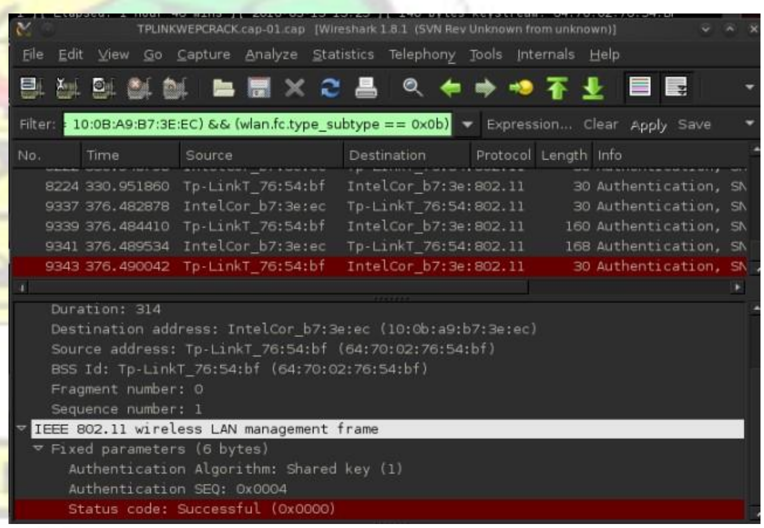


Figure 51: Four captured authentication request and response messages between the legitimate client and the Access Point viewed with Wireshark

3. The "airodump-ng" command was used to capture the packets in figure 50 also saved a copy of keystream byte and the corresponding IV that the legitimate client used to encrypt the challenge plaintext. This is shown as a .xor file as shown in figure 52.

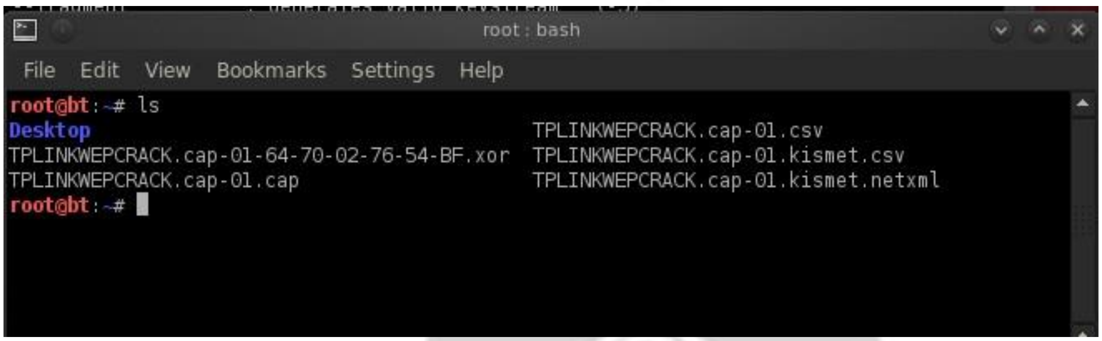
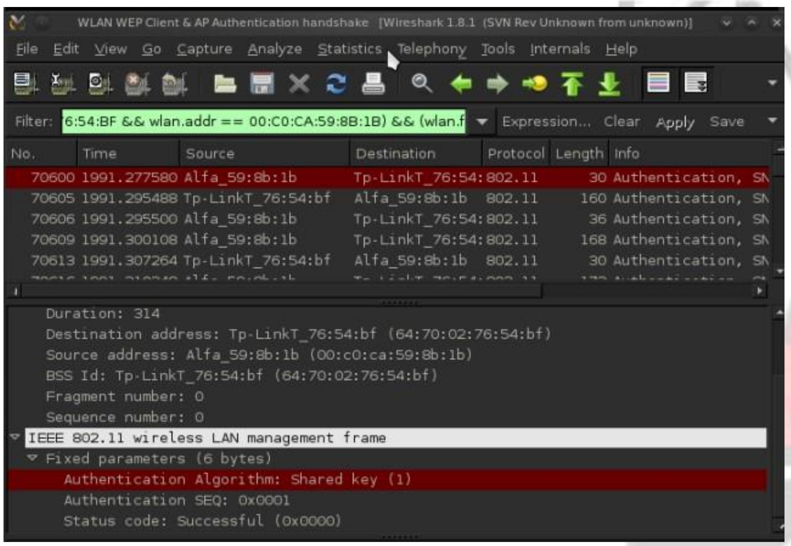


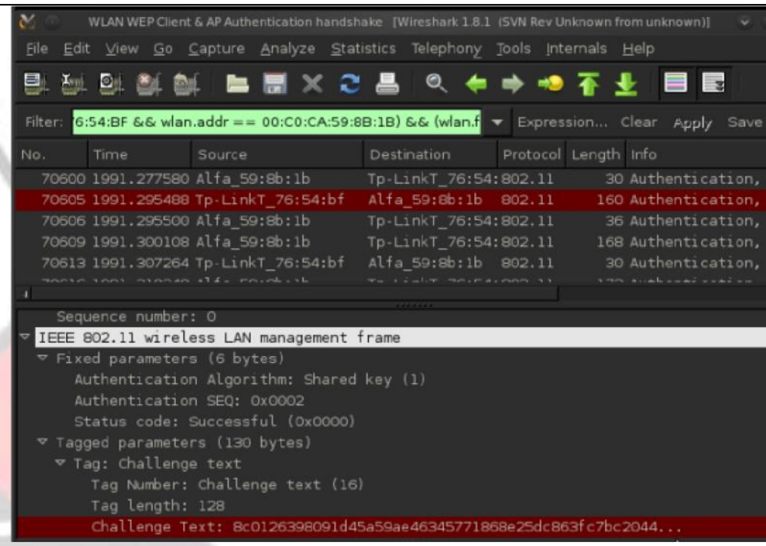
Figure 52: A copy of the keystream byte and corresponding IV saved by airodump-ng

4. The attacker laptop was now used to send an authentication request packet to the legitimate AP as shown by Message 1 in figure 53.
5. The Access point responded to the request with a challenge plaintext message to the attacker's laptop as shown in Message 2 of figure 53.
6. The attacker then responded to the challenge plaintext by encrypting it with the captured keystream byte and corresponding IV as shown in Message 3 of figure 53.
7. The Access Point upon receiving message 3 was able to decrypt it because the keystream byte and IV used by the attacker were correct and that WEP does not avoid keystream or IV reuse.
8. Thus the Access Point granted the authentication access to the attacker as shown in Message 4 of figure 53 even though the attacker does not know the WEP password to the network.

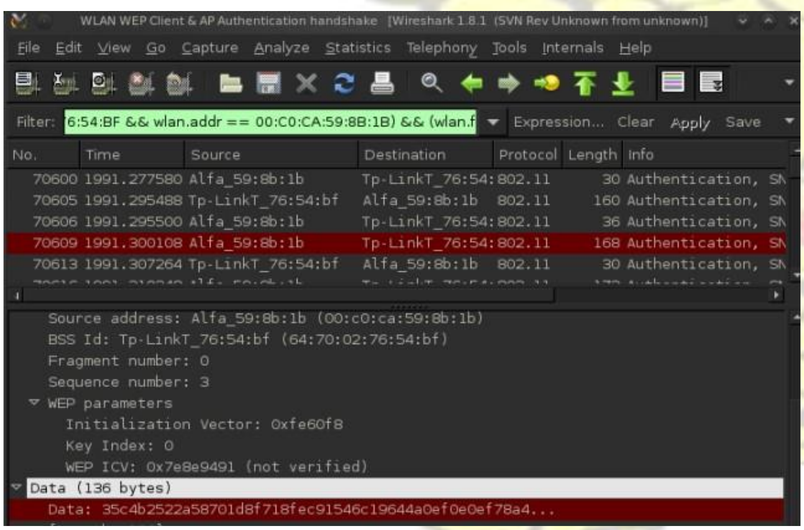
Message 1: **Authentication Request Packet** from hacker to legitimate AP



Message 2: **Challenge Plaintext Packet** from AP to hacker



Message 3: **Challenge Response Ciphertext Packet** from hacker to AP



Message 4: **Authentication Success Packet** from AP to hacker

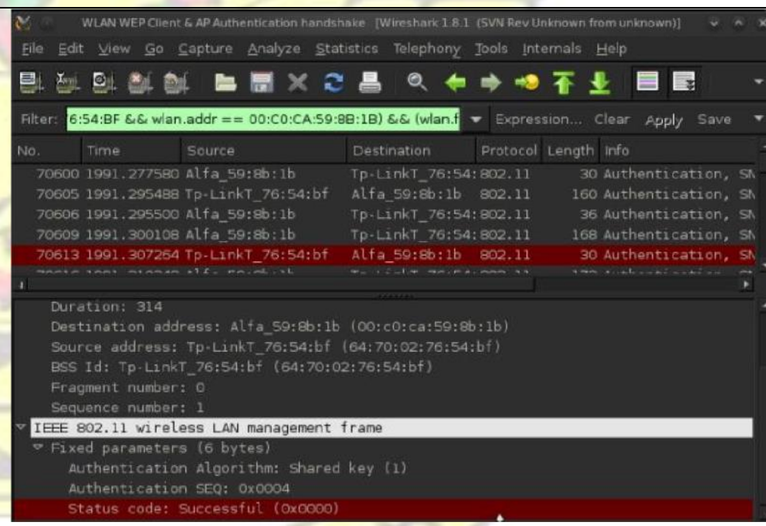


Figure 53: The captured four authentication request and response messages between the hacker and the Access Point viewed with Wireshark

The syntax that was used to execute the above forged authentication into the WEP network is the "aireplay-ng --fakeauth" command as shown in figure 54.

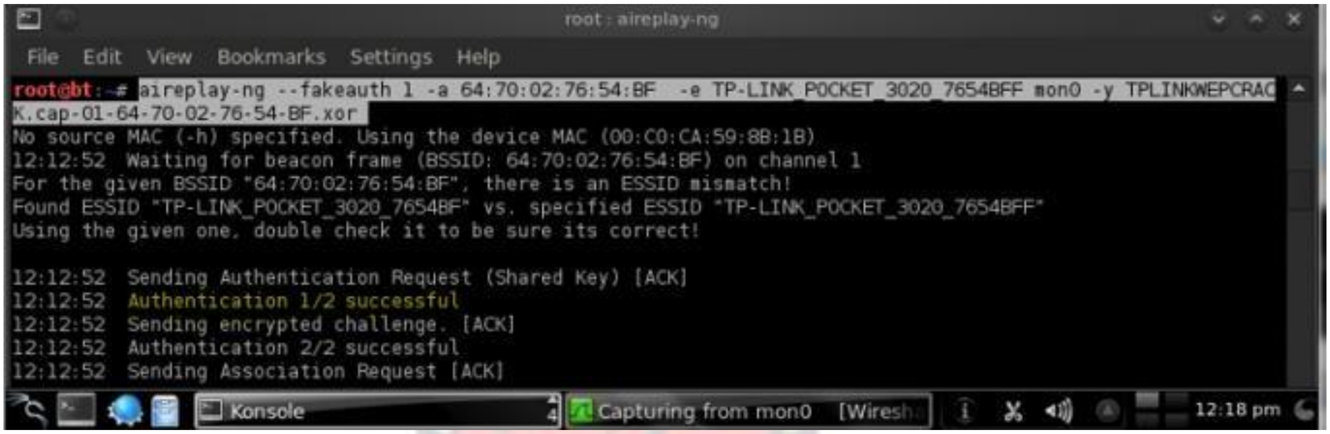


Figure 54: "aireplay-ng –fakeauth" command used to forge authentication into WEP network

The analysis on the success and significance of this experiment is provided in section 4.2

B. WEP DOES NOT SUPPORT MUTUAL AUTHENTICATION

It was proved that there is no mutual authentication in WEP: The access point authenticates the client but the client has no way of proving the authenticity of the access point. Hence a rogue AP can be set-up with the same security properties as the legitimate AP, and legitimate client may connect to this fake AP without knowing that it has connected to the wrong AP. The below experiment provides the evidence of this vulnerability:

1. BackTrack 5 was used to monitor the airwaves for legitimate WEP access points by using the "airodump-ng mon0" command as shown in figure 55:

Figure 55: A legitimate WEP AP with BSSID= 64:70:02:76:54:BF connected to legitimate client with MAC= 10:0B:A9:B73E:EC

2. BackTrack5 was used to bring up a fake soft Access Point by specifying support for WEP (-W 1), channel (-c), bssid (-a), and essid (-e) using the "airbase-ng" command as shown in figure 56.

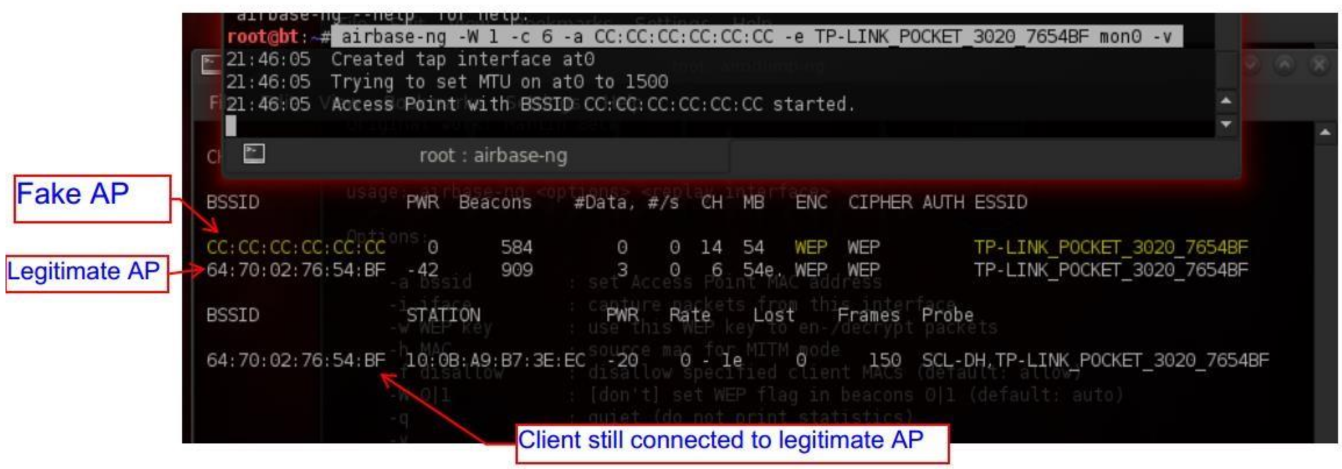


Figure 56: A fake AP powered on with airbase-ng tool in BackTrack5

3. Next "aireplay-ng --deauth" command as shown in figure 57 was used to force the legitimate AP to go offline so that all the legitimate clients will begin to establish reconnection requests.

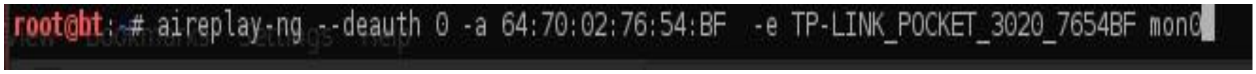


Figure 57: The “deauth” command in BackTrack used to disconnect all clients from the AP

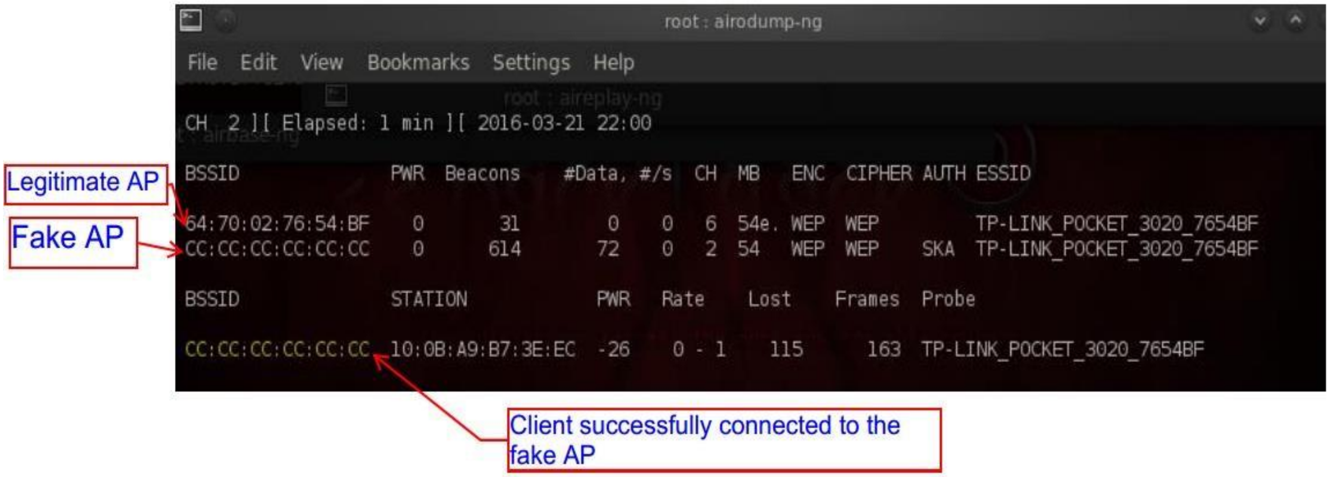
4. A disconnected client in its attempt to re-establish the lost connection, will begin to send probe request messages searching for the WEP network it had previously connected to.
5. The fake AP responded to the probe and the legitimate client sent an authentication request to the fake AP.
6. The fake AP then sent a challenge message to the legitimate client and the client encrypted it with its secret WEP key and responded.
7. Because there is no mutual authentication in WEP, the client had no prove that the network it is connecting to, was the fake one.
8. The fake AP can pretended it was able to check the encrypted sent by the client and responded with an authentication success message to the client without ever knowing the WEP key as shown in figure 58.

Figure 58: The legitimate client successfully connected to the fake AP

The analysis on the success and significance of this experiment is provided in section 4.2

C. VULNERABILITY IN THE USE OF ICV IN WEP WHICH LEADS TO MESSAGE MODIFICATION AND MESSAGE INJECTION ATTACKS IN WEP

It was proved that it is possible to capture a WEP encrypted packet, modify it and inject it back into the network without been detected by the WEP Security protocol as a tempered packet. The below experiment provides evidence of this vulnerability.



1. “aireplay-ng –arpreplay" was used to capture a WEP encrypted ARP request packet from a host as shown in figure 59.

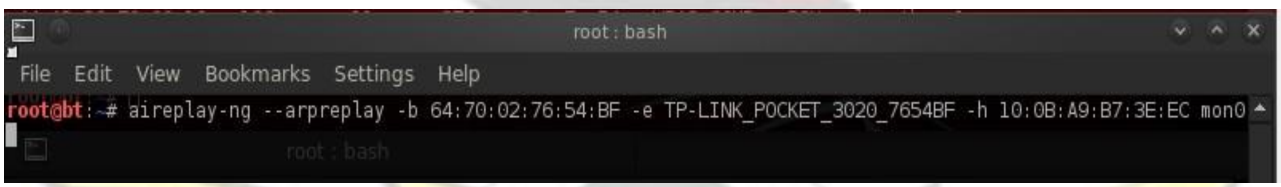


Figure 59: The “aireplay-ng –arpreplay” command

2. “aireplay-ng –arpreplay" modified (bit-flipped) the packet into an ARP request packet for the same host and computed a new ICV for the modified packet.

2. This modified ARP packet was accepted by the WEP network and more ARP packets were generated between the legitmate client and the attacker’s machine as shown in figure 60 because of the way WEP deploys ICV in its encryption mechanism.

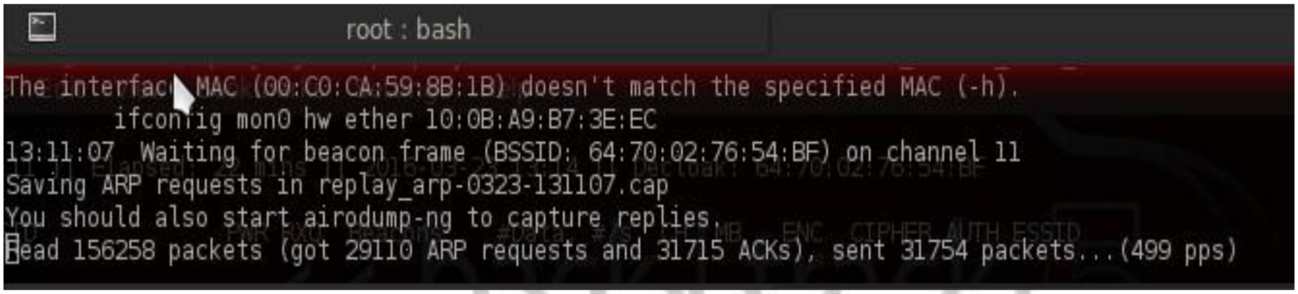


Figure 60: The results of the aireplay-ng --arpreplay command

The analysis on the success and significance of this experiment is provided in section 4.2.

3.4 Cracking IEEE 802.11 WEP Key

With the discovered vulnerabilities in WEP, an attempt was made to crack the password of a WEP secured WLAN. The below experiment provides the evidence of this attempted attack:

1. An Access Point was configured to support WEP security protocol with a password as shown in figure 61. For example, the password “come123@123co” was chosen.

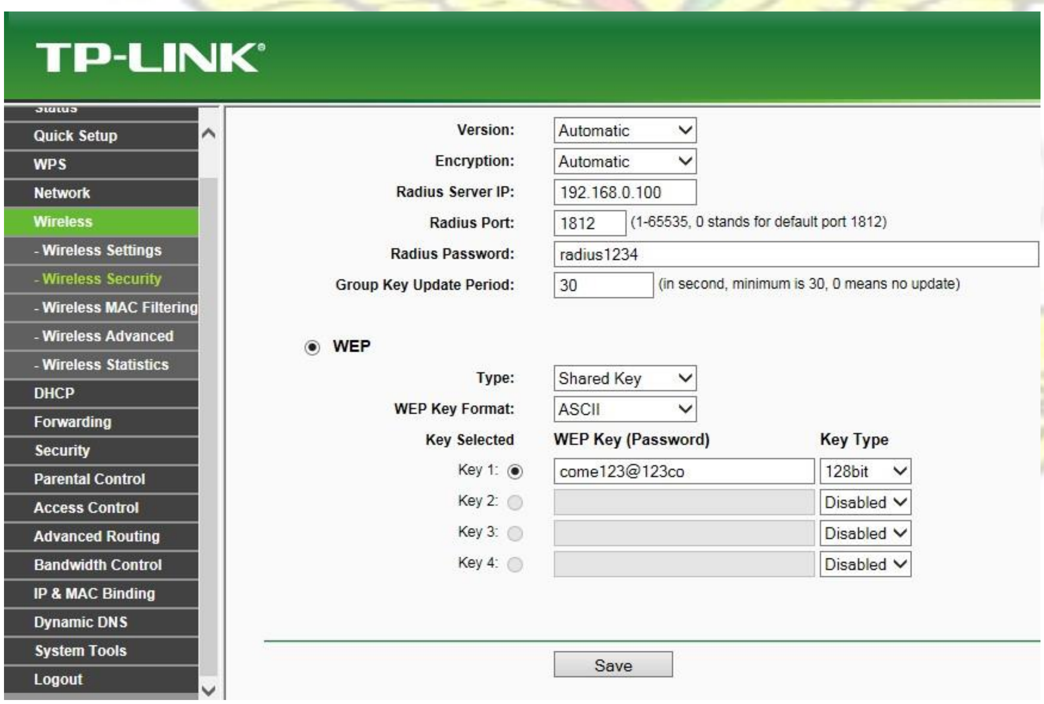


Figure 61: The access point configured to support WEP with password “come123@123co”

2. A legitimate client was also configured to support WEP security protocol with the same password as shown in figure 62. The password “come123@123co” was chosen.

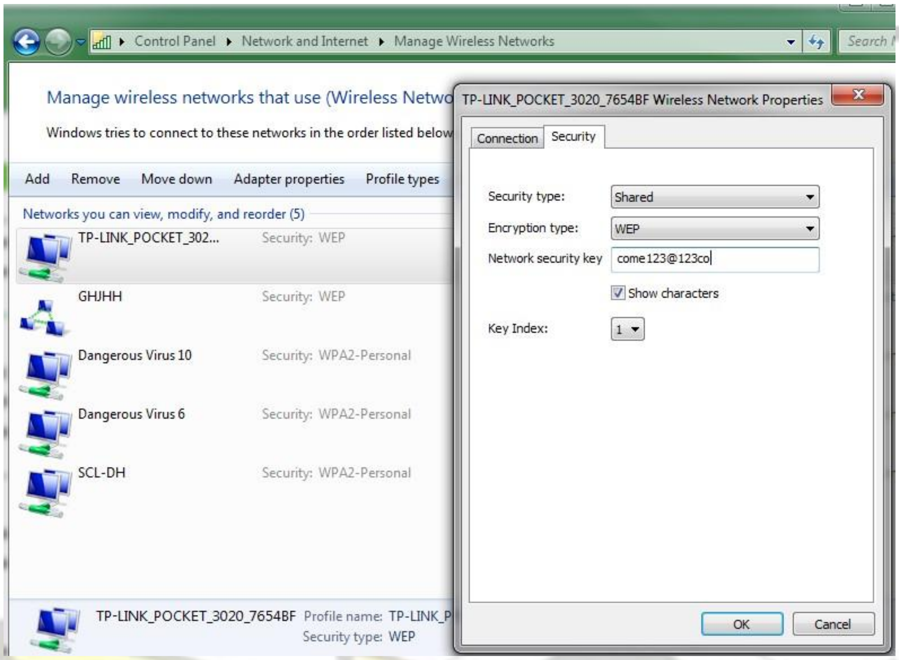


Figure 62: The legitimate client configured to support WEP

3. The client was then connected to the WEP Access Point network as shown in figure 63.

Figure 63: The legitimate client connected to the WEP network

4. Backtrack 5 was used to monitor all the available wireless networks with the command “airodump-ng mon0” as shown in figure 64 and results shown in figure 65. The packets of the WEP network were eavesdrop and written to a file as shown in figure 64.

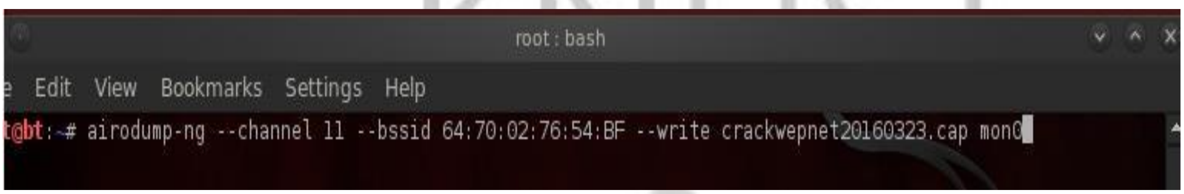


Figure 64: The “airodump-ng” command used capture and save the WEP network packets

Figure 65: The results of the “airodump-ng” command

5. Next, the vulnerability in WEP use of ICV (refer to section 3.3 C) was used to capture an ARP packet, spoofed the mac-address of a legitimate client, modified the ARP packet for the same host, and replayed it back into the network with the “aireplay-ng --arpreplay” command as shown in figure 66.

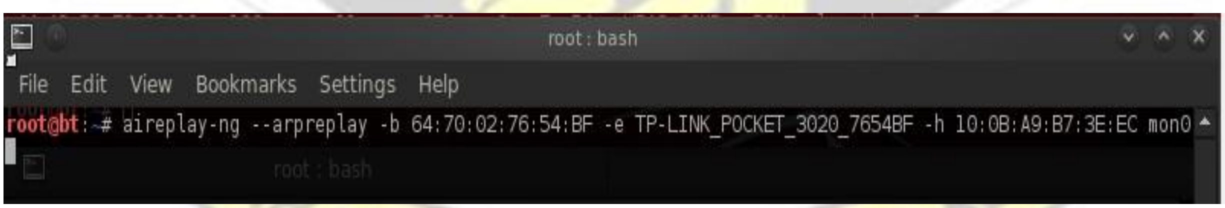


Figure 66: The “aireplay-ng –arpreplay” command

6. However, it failed to capture an ARP packet. So the “aireplay-ng --deauth" command was used to disconnect all the clients from the AP as shown in figure 67 so that clients will reestablish

connection and hence re-generate more ARP packets. The “Ctrl+c” command was used to stop the “deauth” attack after 1 minute.

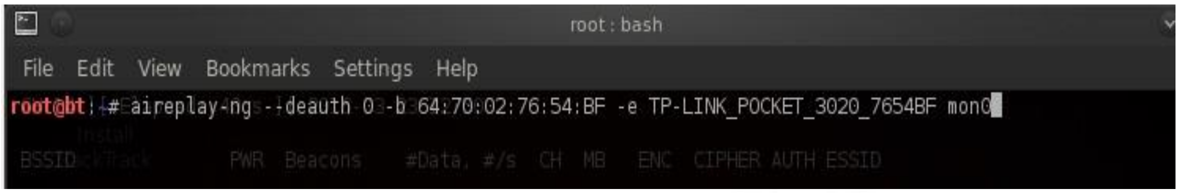


Figure 67: The “aireplay-ng --deauth" command

7. The “aireplay-ng –arpreplay” command was repeated and ARP packets were successfully captured and replayed back into the network as shown in figure 68.

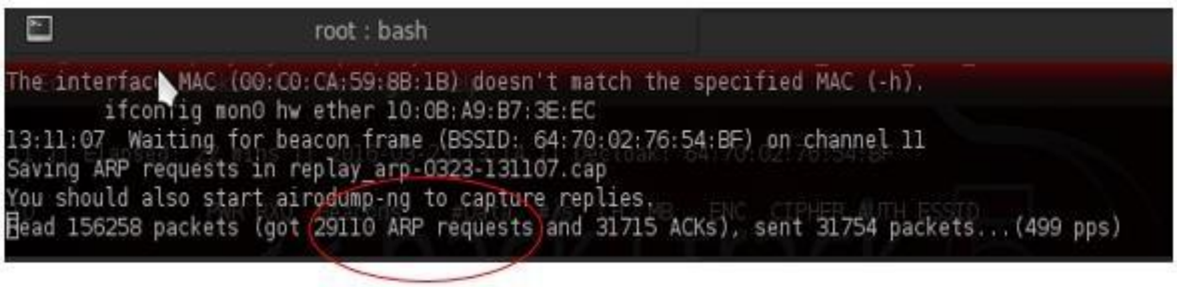


Figure 68: The results of the “aireplay-ng –arpreplay” command

8. The “ls” command was used to locate all the saved .cap file as shown in figure 69.

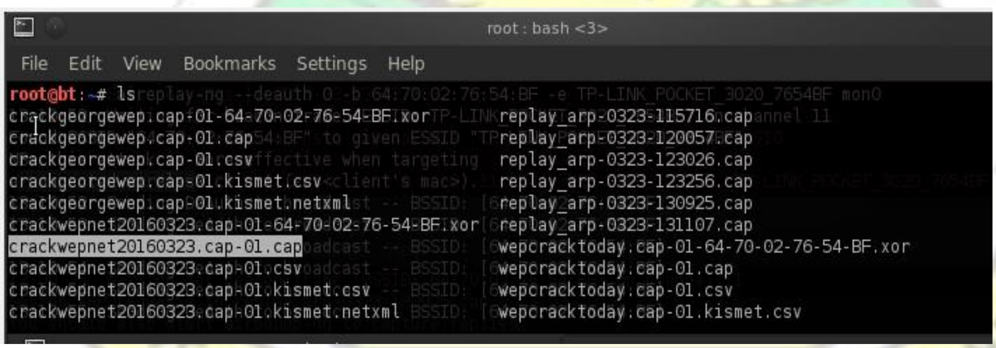


Figure 69: The results of the ls command

9. The more the WEP encrypted ARP packets generated between the legitimate AP and the attacker machine, the more IVs containing weak IVs which have a correlation with the WEP secret key and their corresponding keystream bytes are generated. This vulnerability in weak

IVs and their corresponding keystream bytes were first discovered by (Stubblefield et al, 2002; Kazukuni & Hideki, 2008) and is used in this experiment.

10. The "aircrack-ng" command was used together with the saved .cap file in an attempt to statistically discover the WEP secret key as shown in figure 70.

Figure 70: The aircrack-ng command

11. After capturing 66,560 IVs which have a statistical correlation with the WEP key as show in figure 71, the WEP secret key was cracked in less than 5 minutes as shown in figure 71.

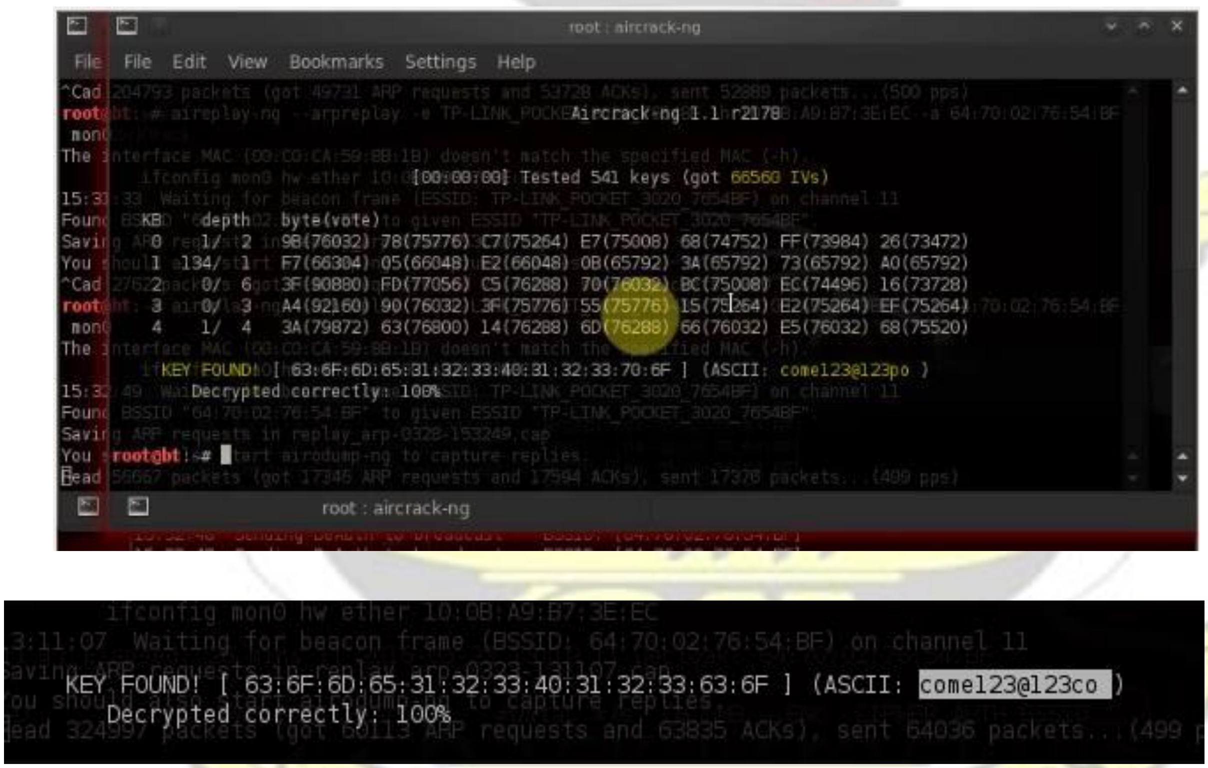


Figure 71: The successful cracking of the WEP Password

The analysis on the success and significance of this experiment is provided in section 4.3

3.5 Vulnerabilities in IEEE 802.11 WPA/ WPA2-PSK Security Protocol

After studying and analysis literature on the architecture of the IEEE 802.11 WPA/ WPA-2 PSK security protocol (refer to section 2.3), the following three vulnerabilities were discovered:

1. **The four-way EAPOL Handshake are not encrypted over the air interface**: They are plaintext. A hacker can eavesdrop on the four way handshake and easily retrieve the ANounce, SNounce, Authenticator Mac Address, Supplicant Mac Address, and the MIC messages as shown in figures 26 to 30.
2. **The formulae for deriving the PMK and PTK are known to any adversary**: The formula for PMK is *PMK = PBKDF2 (PassPhrase, ssid, ssidLength, 4096, 256)* and PTK is *PTK = Function (PMK, ANounce, SNounce, Authenticator Mac Address, and Supplicant Mac Address).*
3. **There is an MIC (Plaintext) to ensure that the computed PTK is the same as the one computed by the legitimate client**: This MIC can be used to verify that the captured PTK and the computed PTK are the same which may lead to a possible use of a dictionary file to brute force the PassPhrase.

With the discovered vulnerabilities in WPA/ WPA2-PSK, an attempt was made to crack the password of a WPA/ WPA2-PSK secured WLAN. The below experiment provides the evidence of this attempted attack:

3.6 Cracking IEEE 802.11 WPA/ WPA-2 PSK Passphrase

1. An Access Point was configured to support WPA/ WPA-2 PSK as shown in figure 72: The WPA/ WPA-2 PSK passphrase was “Ashes@112”.

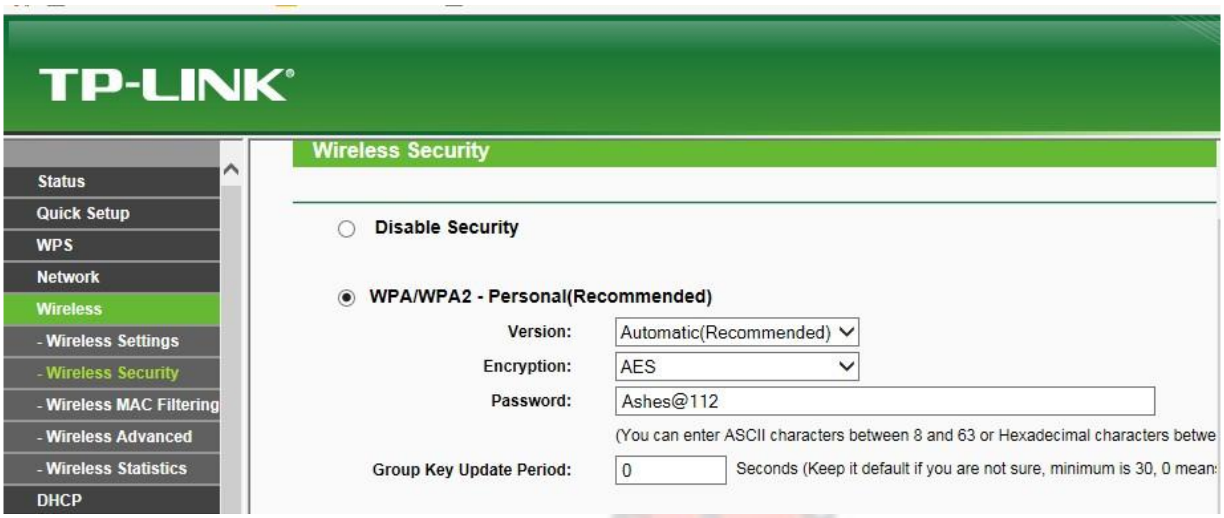


Figure 72: The AP configured to support WPA/ WPA-2 PSK with password “Ashes@112”

2. A legitimate client was also configured to support the WPA/ WPA-2 PSK with the same security credentials as the access point as shown in figure 73. The client then connected to the access point.

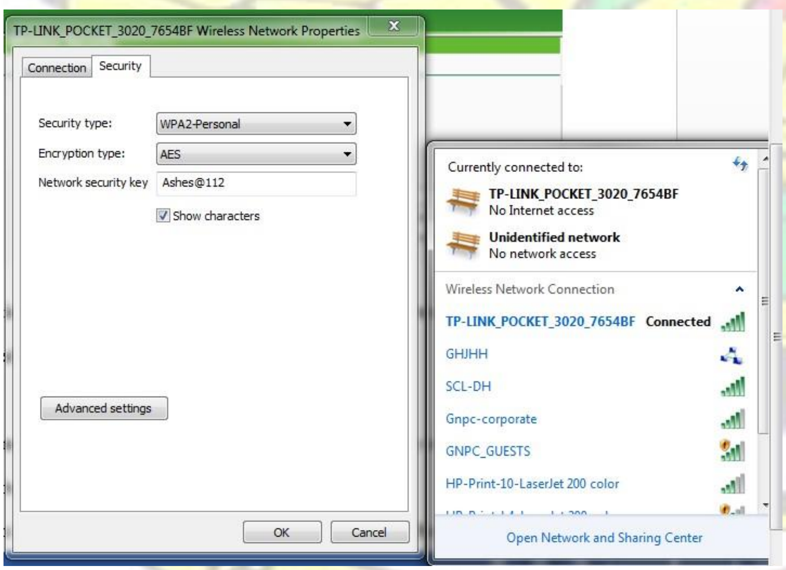


Figure 73: The legitimate client configured to support WPA/ WPA-2 PSK with same password as the AP

3. “airodump-ng mon0” command was used to monitor the WPA/ WPA2-PSK network called

TP-LINK_POCKET_3020_7654BF as shown in figure 74. It cipher suite was CCMP.

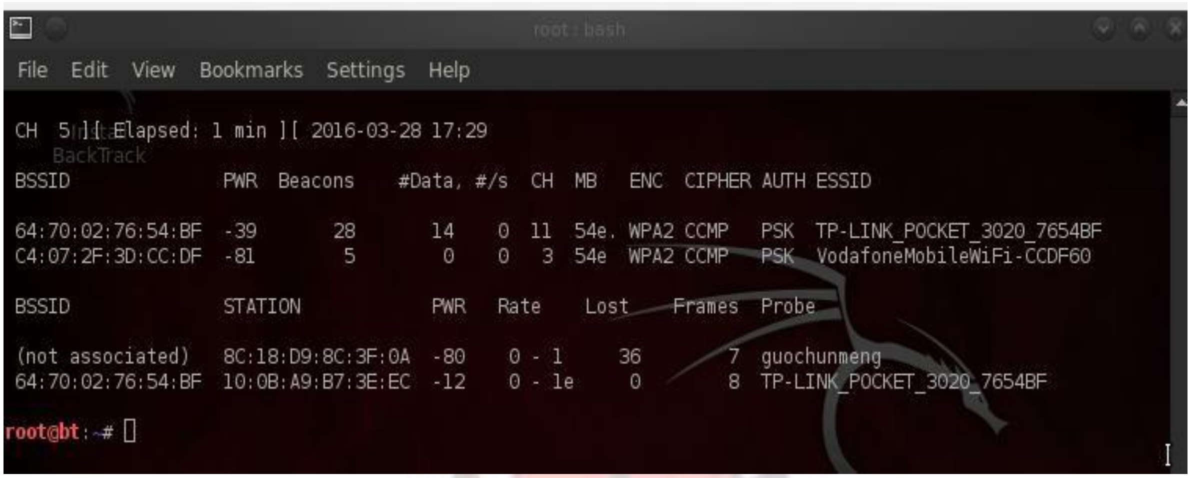


Figure 74: The output of “airodump-ng” used to monitor broadcasting SSIDs

4. Next the four-way EAPOL handshake between the AP and the legitimate client were captured and saved to a .pcap file as shown in figure 75:

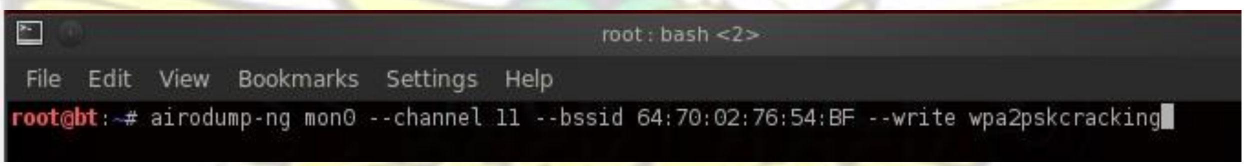


Figure 75: The eavesdropping and saving of the PSK EAPOL Handshake

5. As soon as a legitimate client re-connected to the AP, “airodump-ng” prompted that it has successfully captured the four-way EAPOL handshake as shown in figure 76.

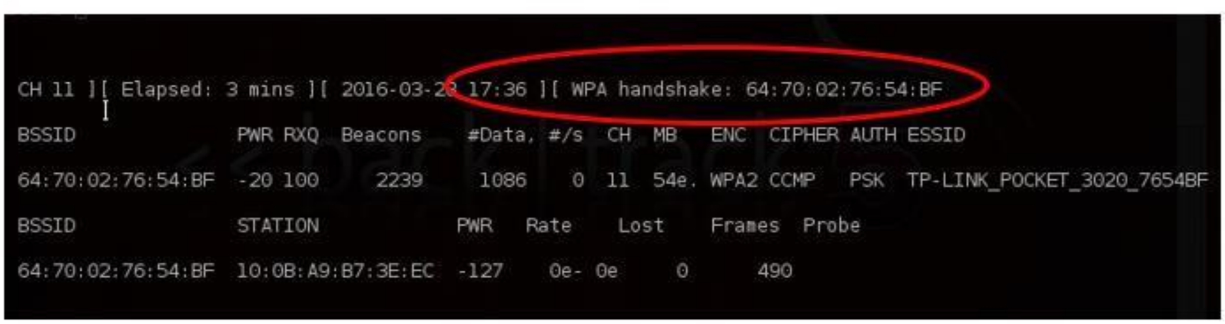


Figure 76: The capturing of the four-way EAPOL handshake as soon as a legitimate client connects to the legitimate AP

6. Next a default dictionary file directory in Backtrack5 was opened and edited to include additional potential passwords as shown in figures 77 and 78.

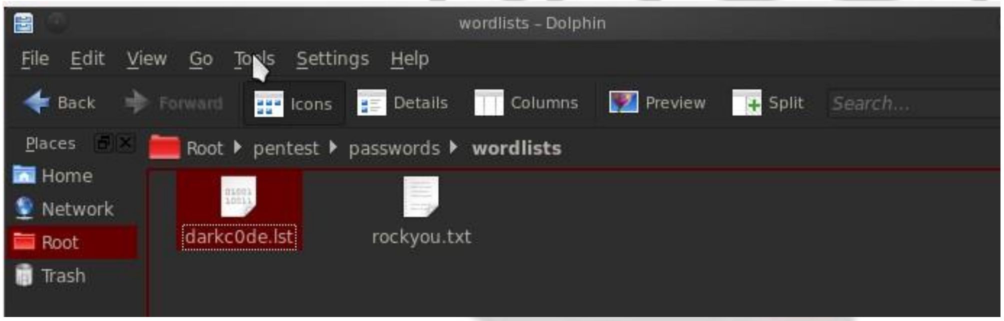


Figure 77: The location of a default dictionary file in BackTrack5

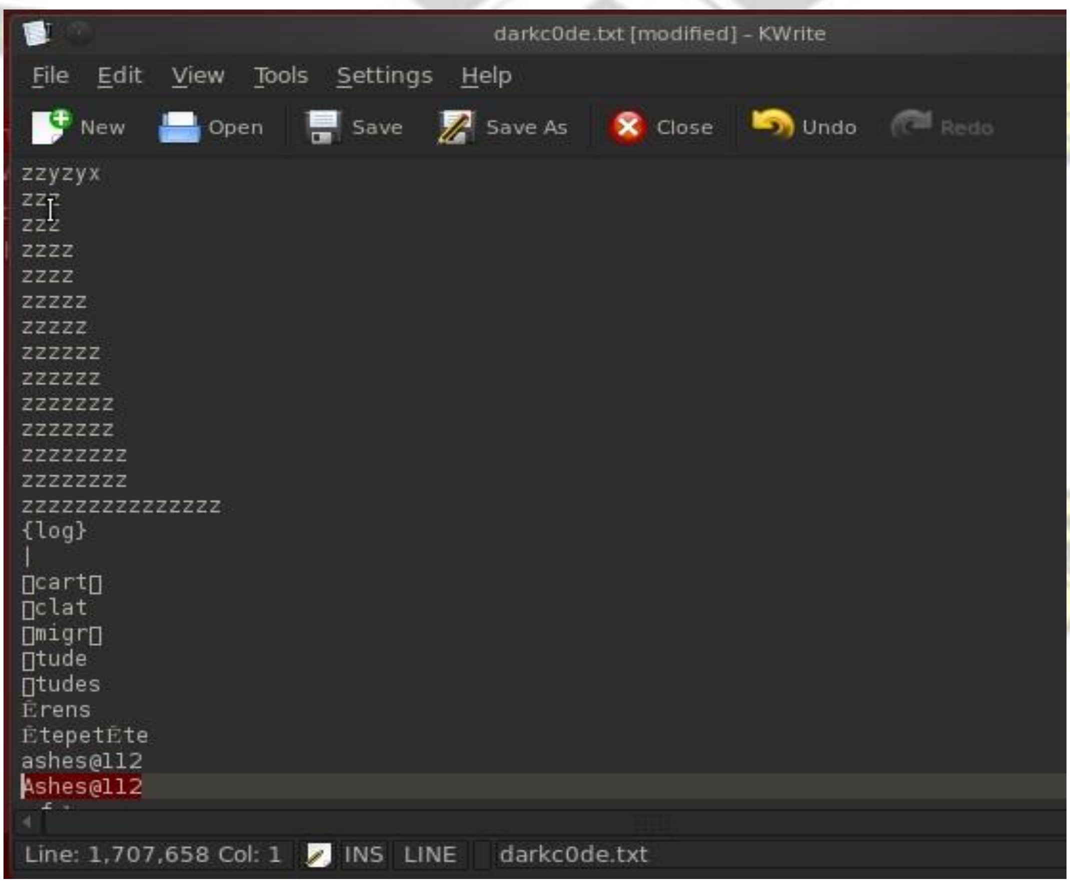


Figure 78: The contents of the default dictionary file edited to include “Ashes@112” passwords

7. Next “aircrack-ng” command was used together with the captured EAPOL handshake file , and a link to the dictionary file in an attempt crack the WPA/ WPA-2 PSK password as shown in figure 79:

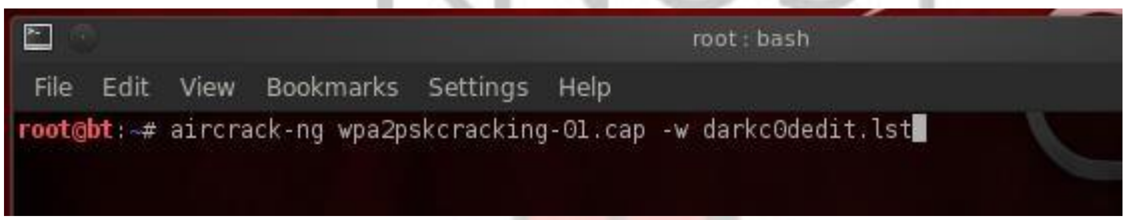


Figure 79: “aircrack-ng” fed with the captured four-way handshake file and a dictionary file

8. “Aircrack-ng” used the dictionary file and tried various combinations of passphrases to bruteforce the password. For each guessed passphrase, it computed the PMK (Master key), PTK (Transient key), and the MIC (EAPOL HMAC) as shown in figure 80. It then compared the computed MIC with the captured MIC. If there was a match, it knew that the chosen passphrase was correct otherwise another passphrase was chosen and the attack repeated.

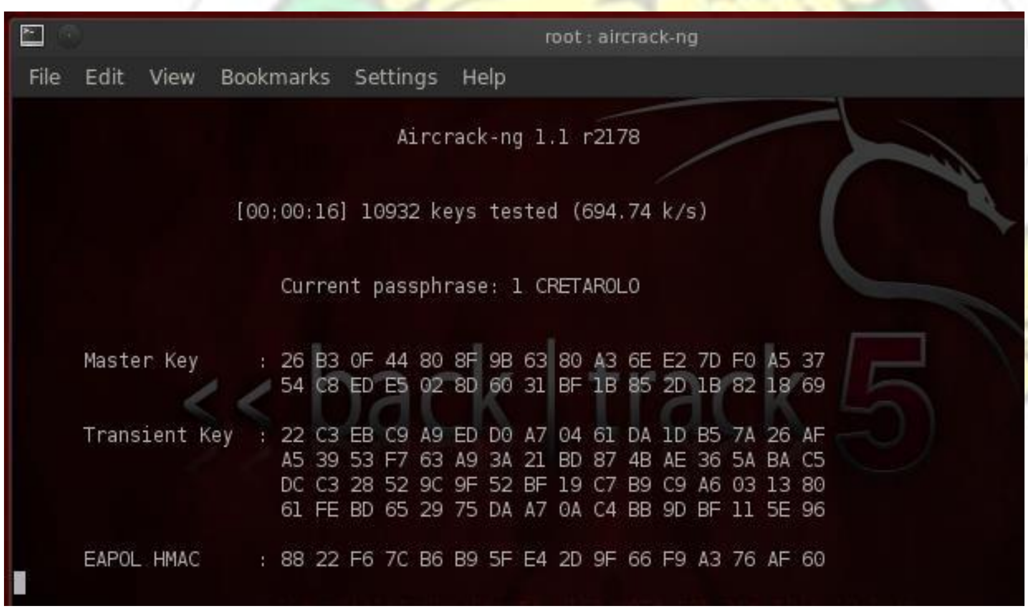


Figure 80: “Aircrack-ng” trying various combinations of passphrase in an attempt to crack the key

9. Because the password to the WPA/ WPA-2 PSK network was in the attacker's dictionary, it was successfully cracked after testing 1,144,845 passwords in the attacker's dictionary as shown in figure 81.

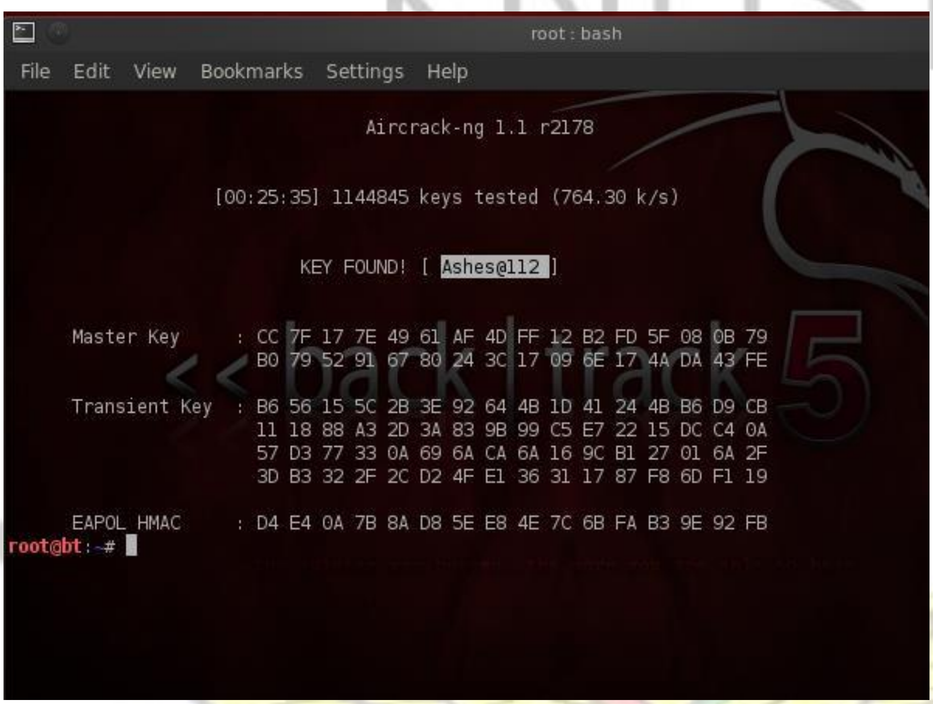


Figure 81: "Aircrack-ng" locating the correct passphrase and hence cracking the WPA/ WPA-2 PSK key

The analysis on the success and significance of this experiment is provided in section 4.5

3.7 Vulnerabilities in IEEE 802.11 WPA/ WPA2-EAP MD5 Security Protocol

After studying and analysis literature on the architecture of the IEEE 802.11 WPA/ WPA-2 EAP MD5 security protocol (refer to section 2.4.4), the following four vulnerabilities were discovered:

1. **EAP-MD5 does not provide support for confidentiality**: All the messages including the EAP-Message ID, RADIUS challenge value, and the EAP MD5 Hash value are all sent in plaintext between the Supplicant and the Authentication server.

2. **The formula for computing the MD5 Hash function is known to any adversary**: The *EAP MD5 Hash value = (EAP-Message ID + User Password + RADIUS challenge value)*. Hence an attacker who eavesdrops on these packets may be able to guess the user password by computing the MD5 hash for each guessed password and comparing it with the captured MD5 Hash value.

3. **EAP-MD5 does not support mutual authentication**: It is the supplicant who proves its identity to the Authentication server through the response to the MD5 Challenge message. The Authentication server does not prove its identity to the supplicant. Hence as proved for WEP in section 3.3 B, any rogue server can to pretend that it was able to prove the identity of the supplicant and send an EAP Success message to the supplicant.

4. **EAP-MD5 does not support any key generation mechanism like RC4, AES, 3DES, and etc**: User/Authenticator passwords are stored in plaintext within the server.

With the discovered vulnerabilities in WPA/ WPA2-EAP MD5, an attempt was made to crack the password of an WPA/ WPA2-EAP MD5 secured WLAN. The below experiment provides the evidence of this attempted attack:

3.8 Cracking IEEE 802.11 WPA/WPA2-EAP MD5 Passphrase

1. "airodump-ng" command was used to monitor a WPA/ WPA2-EAP network called TP-LINK_POCKET_3020_7654BF as shown in figure 82.

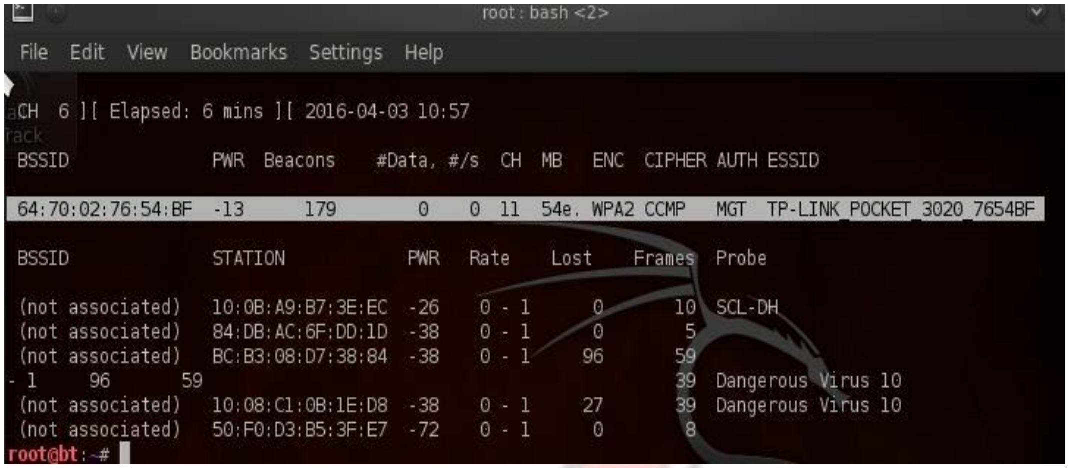


Figure 82: The results of “airodump-ng” to monitor a WPA/ WPA2-EAP WLANs

2. “airodump-ng --write" command was used to save the EAP EAPOL messages between the legitimate server and the legitimate supplicant.
3. “EAPMD5crack” software in BackTrack5 was supplied with the captured EAP EAPOL pcap file, and a dictionary file to attempt cracking the MD5 key as shown in figure 83.

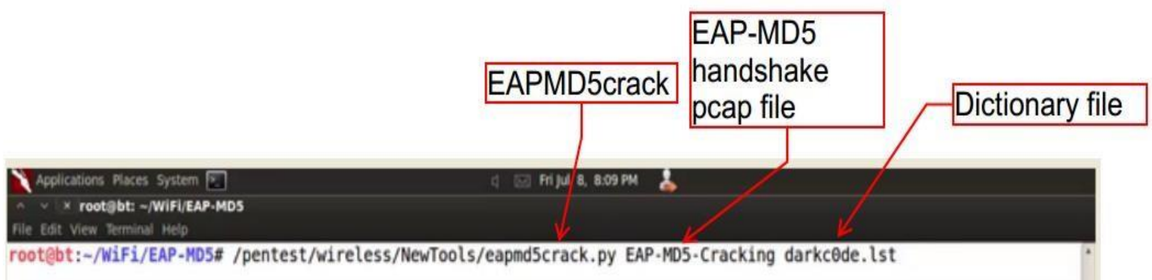


Figure 83: The “EAPMD5crack” command

4. The EAPMD5crack software read the captured file and detected the EAP-MD5 challenge, EAP Response ID, and the MD5 hash within the pcap file as shown in figure 84.

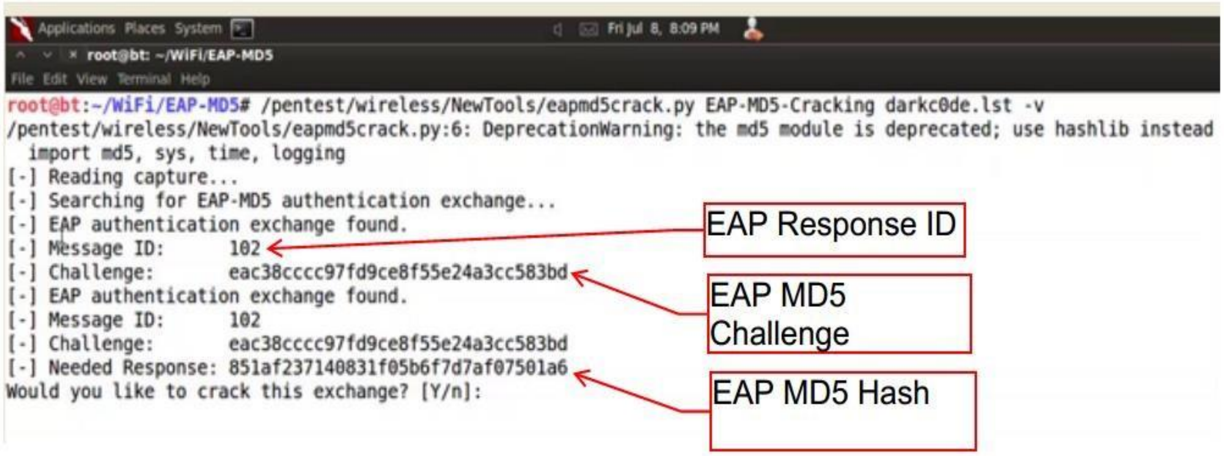


Figure 84: EAPMD5crack detecting an EAP Response ID, MD5 Challenge and MD5 Hash in an EAP-MD5 Authentication exchange packets

5. The "EAPMD5crack" software begun attempting various passwords within the dictionary file to crack the MD5 password as shown in figure 85. It computed the MD5 Hash for each guessed password and compared it with the captured MD5 Hash to see if there was a match.

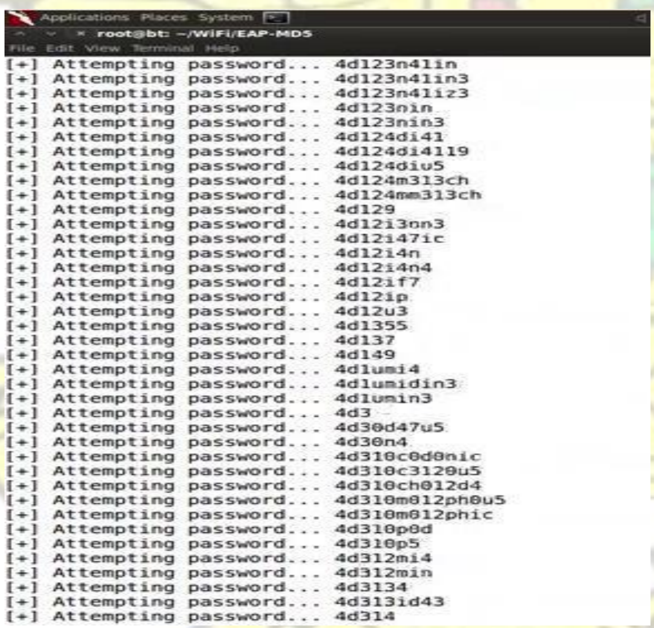


Figure 85: EAPMD5crack attempting a number of passwords in the dictionary file

6. Because the password was in the dictionary file, there was a match with the MD5 Hash and hence the EAP MD5 password was successfully cracked as shown in figure 86.

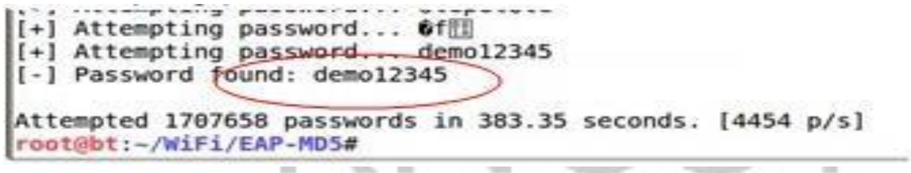


Figure 86: EAPMD5crack cracking the EAP-MD5 client-server password

The analysis on the success and significance of this experiment is provided in section 4.4

3.9 Vulnerabilities in all IEEE 802.11 WPA/WPA2 EAP Authentication Methods that support the use of only Server-Side digital Certificates (EAP-TTLS & PEAPv0)

After studying and analysis literature on the architecture of the IEEE 802.11 WPA/ WPA-2 EAP Authentication methods that mandates the use of only server-side digital certificates (refer to section 2.4.5), the following three vulnerabilities were discovered:

1. **There is no mandatory support for client-side digital certificates**: The client has no way of validating the legitimacy of the server it is connecting to. Hence it may be possible to setup a rogue server to issue fake server-side certificate which will not be validated by the client.
2. **Server-side digital certificates are not well implemented in WLANs**. Instead of the server encrypting the packet with its private key, and re-encrypting it with the public key of the destination station, it ignores the latter. This is because it is computationally expensive for the server to implement the latter. Hence any station in possession of the public key of the server can decrypt a message that is not meant for it.

3. **The formula for computing the MSCHAPv2 Hash is known to any adversary**: The *MSCHAPv2 Hash value = (EAP-Message ID + User Password (from the user database) + RADIUS Challenge value)*. Hence an attacker who eavesdrops on these packets may be able to guess the user password by computing the MSCHAPv2 hash for each guessed password and comparing it with the captured MSCHAPv2 Hash value.

With the discovered vulnerabilities in the way IEEE 802.11 WPA/ WPA2-EAP implements digital certificates, an attempt was made to crack the password to a WPA/ WPA2-EAP network that uses a server-side digital certificate. The below experiment provides the evidence of this attempted attack:

3.10 Cracking all WPA/WPA2 EAP Authentication Methods that support the use of only Server-Side digital Certificates (EAP-TTLS & PEAPv0)

1. A WLAN was set up involving the supplicant, Authenticator, and Authentication server as shown in figure 87 and refer to appendix B for the setup guide.

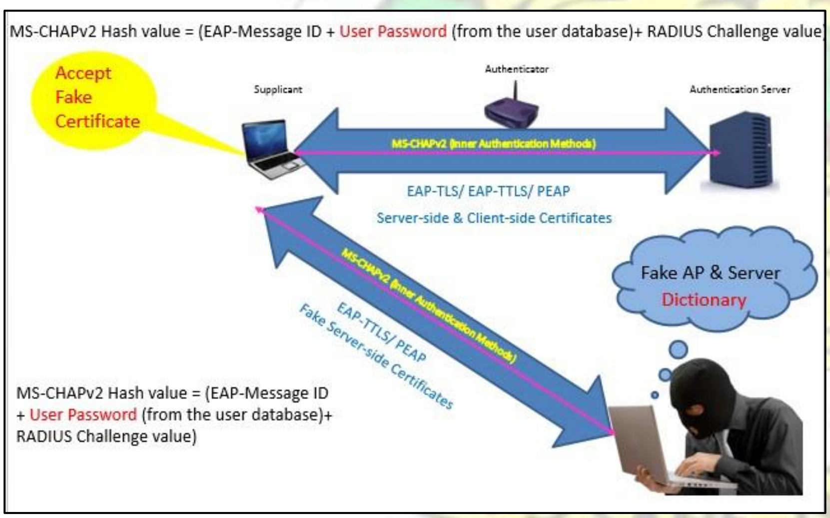


Figure 87: The lab setup to cracking EAP-TTLS or PEAPv0 WPA/ WPA2-EAP networks

2. A fake access point was setup was setup and connected to the freeradius-wpe server in BackTrack 5. Please refer to appendex B for the setup guide.

3. The “eap.conf file” in the Freeradius server was configured to support PEAPv0 with inner authentication method as MS-CHAPv2 as shown in figure 88.

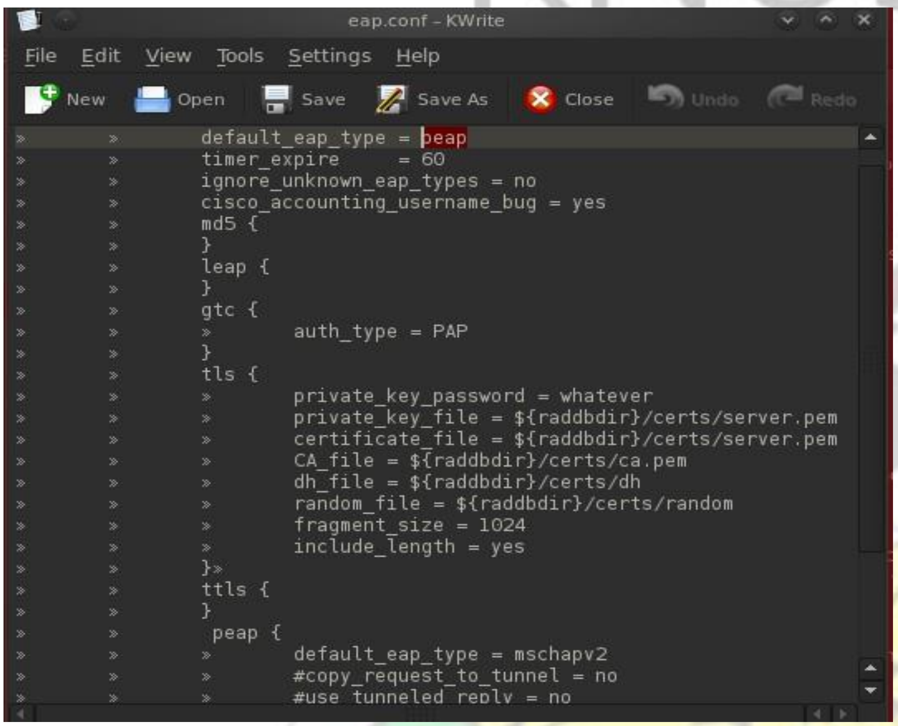


Figure 88: The FreeRadius server configured to support PEAP with MSCHAPv2 as the inner authentication algorithm

4. The freeradius-wpe server was configured to be issue fake digital certificate by using the “./bootstrap” command as shown in figure 89

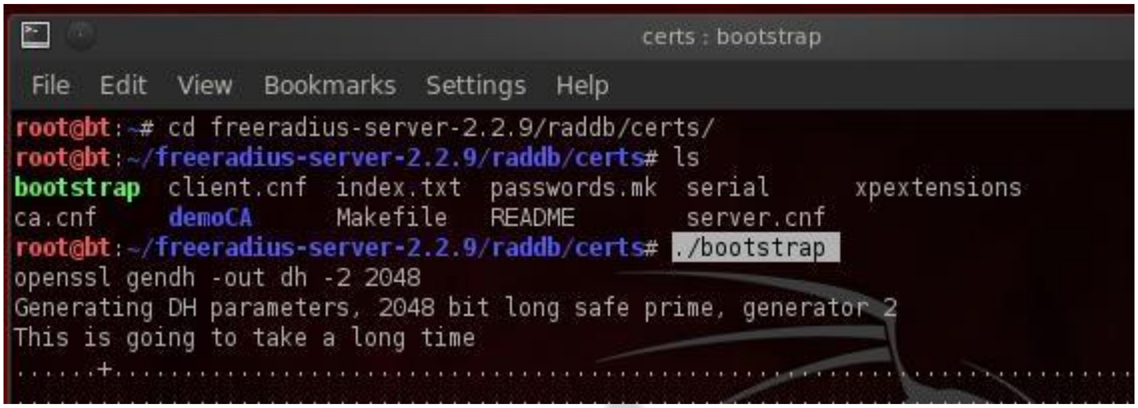


Figure 89: The freeradius fake server-side digital certificate called bootstrap

5. The "tail -f" command was used to log all the transactions between this fake freeradius-server and the legitimate supplicant as shown in figure 90. This is to log all the MSCHAPv2 handshake once the legitimate client accepts the fake digital certificate.

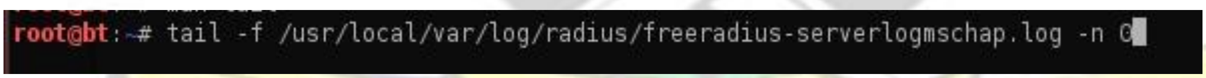


Figure 90: The command to log the MSCHAPv2 handshake into a file.log format

6. The freeradius-wpe server was started with the "radius –X" command as shown in figure 91.

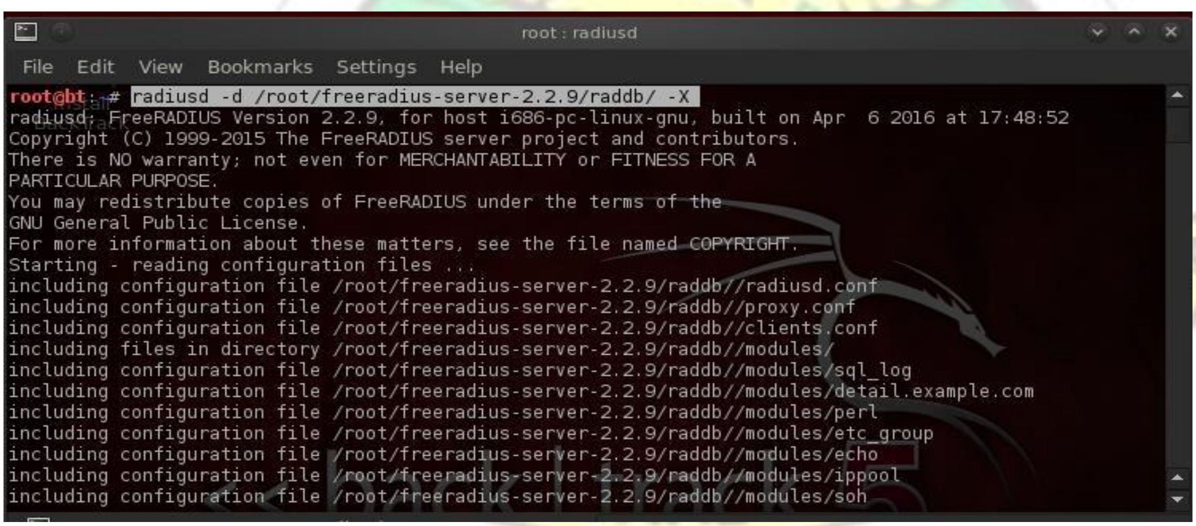


Figure 91: The command for starting the freeradius server

7. As soon as a legitimate client connected to the fake AP, the client was presented with a fake certificate as shown in figure 92.

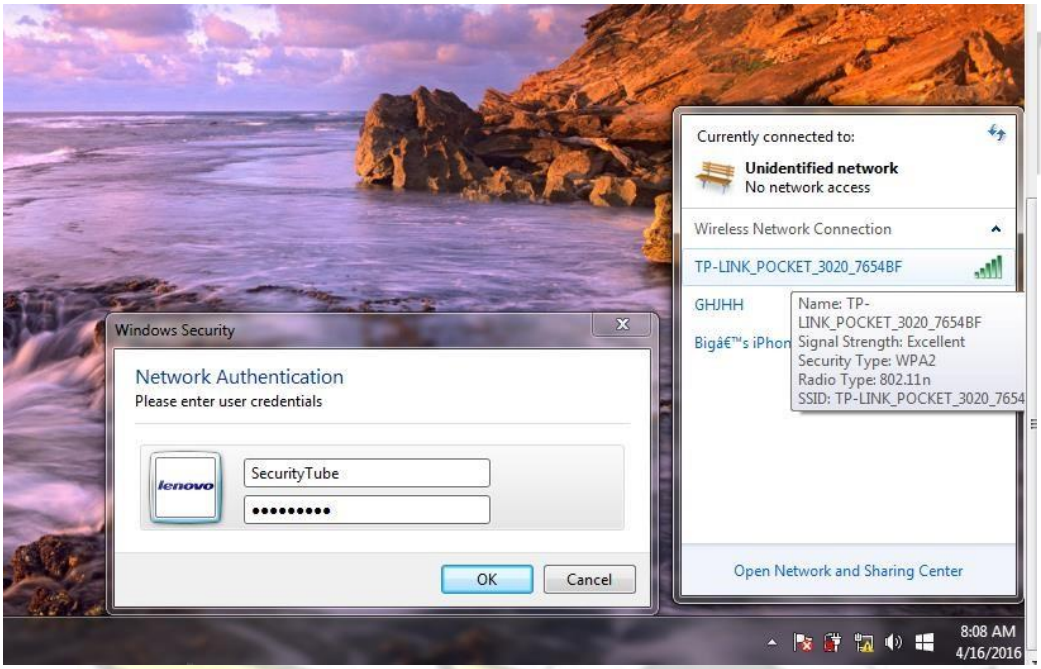
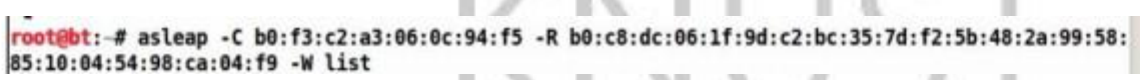


Figure 92: The legitimate client prompted with fake digital certificate to supply security credentials

8. As soon as the legitimate client fails to validate the source of this certificate and inputs the correct username/ password into this fake certificate, the log file will capture the MSCHAPv2 handshake as shown in figure 93.

Figure 93: The captured MSCHAPv2 handshake

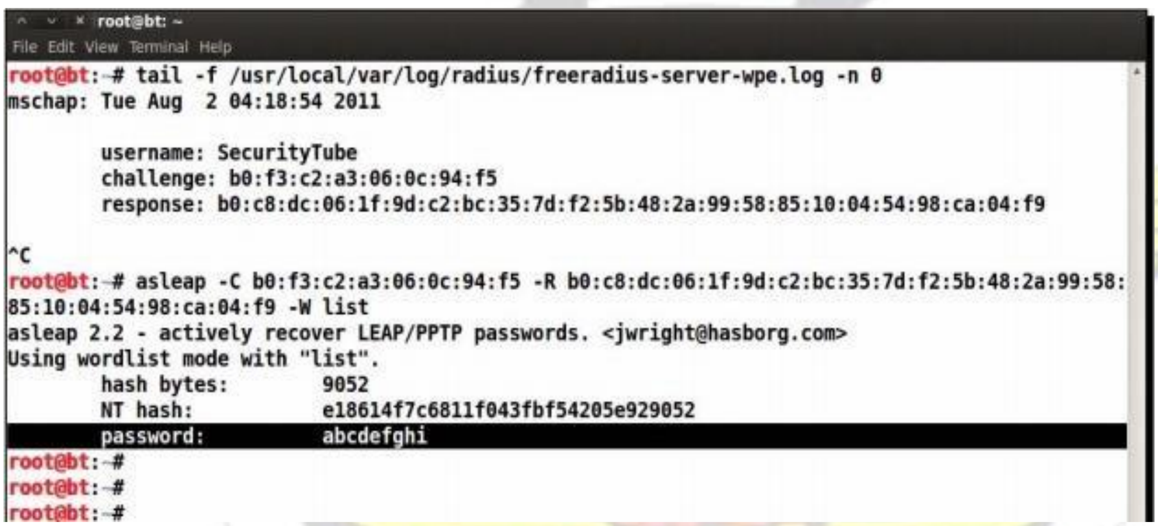
9. “Asleep” software in BackTrack 5 was used to crack the password by supplying it with the captured challenge message (-C), the response to the challenge (-R), and a dictionary file (-W) as shown in figure 94.



```
root@bt:~# asleap -C b0:f3:c2:a3:06:0c:94:f5 -R b0:c8:dc:06:1f:9d:c2:bc:35:7d:f2:5b:48:2a:99:58:
85:10:04:54:98:ca:04:f9 -W list
```

Figure 94: The “asleep” command to crack the MSCHAPv2 hash

10. Asleep software computed the MS-CHAPv2 hash for every guessed password and compared it with the captured MS-CHAPv2 response. Because the password was found within the attacker’s dictionary, the password was successfully cracked as shown in figure 95.



```
root@bt: ~
File Edit View Terminal Help
root@bt:~# tail -f /usr/local/var/log/radius/freeradius-server-wpe.log -n 0
mschap: Tue Aug  2 04:18:54 2011

        username: SecurityTube
        challenge: b0:f3:c2:a3:06:0c:94:f5
        response: b0:c8:dc:06:1f:9d:c2:bc:35:7d:f2:5b:48:2a:99:58:85:10:04:54:98:ca:04:f9

^C
root@bt:~# asleap -C b0:f3:c2:a3:06:0c:94:f5 -R b0:c8:dc:06:1f:9d:c2:bc:35:7d:f2:5b:48:2a:99:58:
85:10:04:54:98:ca:04:f9 -W list
asleap 2.2 - actively recover LEAP/PPTP passwords. <jwright@hasborg.com>
Using wordlist mode with "list".
        hash bytes:        9052
        NT hash:           e18614f7c6811f043fbf54205e929052
        password:          abcdefghi
root@bt:~#
root@bt:~#
root@bt:~#
```

Figure 95: The MSCHAPv2 Password cracked using the “asleep” software

The analysis on the success and significance of this experiment is provided in section 4.9

3.11 Research Survey

The researcher conducted a study by war-driving through some principal streets in Ghana to gather data about the various IEEE 802.11 security protocols that have been configured on WLANs within the country.

“Wigle-Wifi”, a free software on android market as shown in figure 96 was used for the wardriving.

Figure 96: Wigle-Wifi downloaded from Android market

It was installed on a mobile phone as shown in figure 97.

Figure 97: Wigle-Wifi installed on a mobile phone

The researcher drove through most of the principal streets in Accra with the phone in the pocket, to pick up radio signals from Access Points which were broadcasting their SSIDs as shown in figure 98.

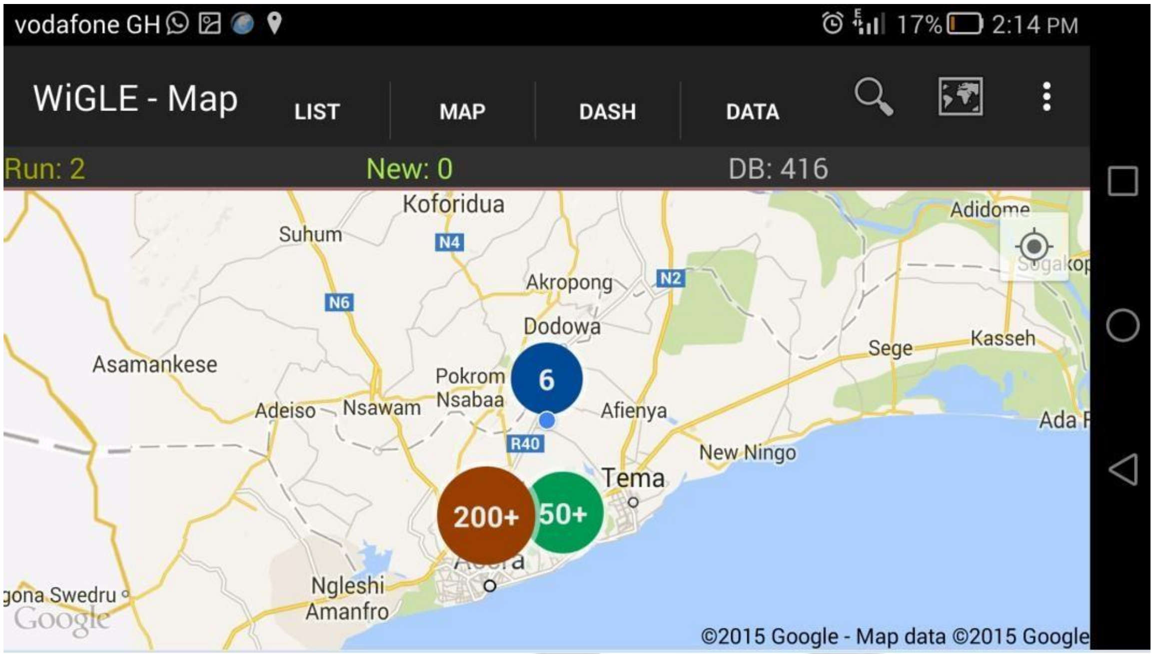


Figure 98: Wigle-Wifi picking WLAN signals with GPS coordinates

The signals included information about the MAC-Addresses of Access Points, Service Set Identifiers (SSIDs), security credentials, channels, signal strengths in dBm, latitude and longitude coordinates of the Access Points. The data is shown in Appendix C. The analysis of the data is done in section 4.5.

CHAPTER 4 ANALYSIS OF THE EXPERIMENTS AND RESEARCH SURVEY

4.1 Overview

This chapter is on analysis. It provides analysis and significance of all the vulnerabilities and successful attacks on IEEE 802.11 security protocols that were discovered in Chapter 3.

4.2 Analysis of the vulnerabilities in WEP

WEP Authentication was successfully compromised: This is because WEP uses an XOR operation to exchange authentication packets between a client and an access point. The XOR exhibits the associative and distributive properties of mathematics: a xor b = b xor a; and (a xor b) xor c = a xor (b xor c). Hence, by performing xor of a copy of the plaintext challenge message with a copy of the encrypted challenge response message, a copy of the keystream byte that was used to encrypt the challenge response message is obtained. This keystream byte was used to forge an authentication into the WEP network and the WEP network granted us success.

The significance of this outcome is that any attacker can eavesdrop on the authentication challenge and response messages, compute the corresponding keystream byte and successfully authenticate to the network without the WEP password.

A client was successfully lured to authenticate to a fake WEP AP: Because there is no mutual authentication in WEP, the client successfully accepted the access and begun sending encrypted WEP packets to the fake AP.

The significance of this outcome is that an attacker can collect all the encrypted WEP packets from a client. These encrypted WEP packets can be saved for later offline statistical attacks to retrieve the WEP password without the presents of the legitimate access point.

A WEP encrypted packet was successfully captured, modified, and injected back into the network without detection: This is because WEP uses an ICV which is linear and mathematically distributive: a xor (b xor c) = (a xor b) xor (a xor c).

The reason for the success is explained as follows:

Let C be a ciphertext intercepted by an attacker. Let's assume that C corresponds to some unknown message M as shown in figure 99.

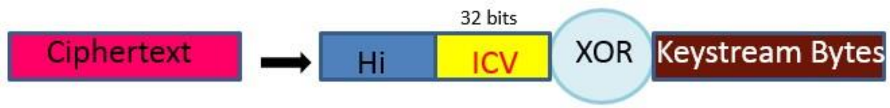


Figure 99: A captured WEP encrypted packet

Now an attacker can create an arbitrary bit-mask (Δ) of the same size as the encrypted data. The attacker can then compute a CRC-32 checksum for this bit-mask (c(Δ)) as shown in figure 100.

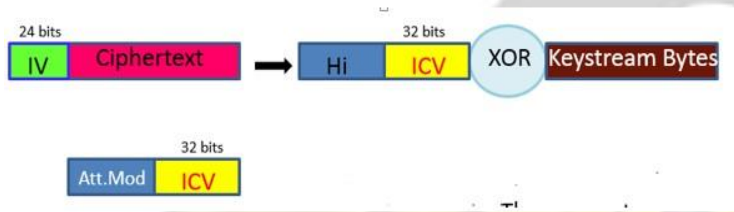


Figure 100: A modified bitmask packet with its computed ICV

This bitmask (Δ,c(Δ)) can be XORed with the original Ciphertext (C) to produce a new Ciphertext (C′) which is the modified WEP encrypted packet as shown in figure 101.

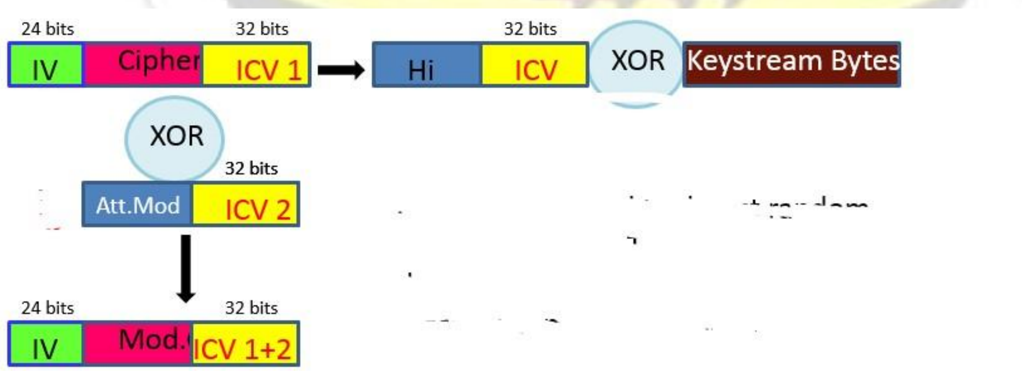


Figure 101: The successfully modified WEP encrypted packet with a new computed ICV

The significance of this outcome is that an attacker can modify a WEP encrypted packet without knowledge of the WEP password. This led us to successfully crack the WEP key.

4.3 Analysis of the Success of cracking WEP Passwords

The WEP Key is either 64-bit or 128-bit long. Thus there are 8 key-bytes for a 64-bit WEP key and 16 key-bytes for a 128-bit WEP key as shown in figure 102. The first 3 key-bytes are the IVs which are known because they are always sent in clear text.

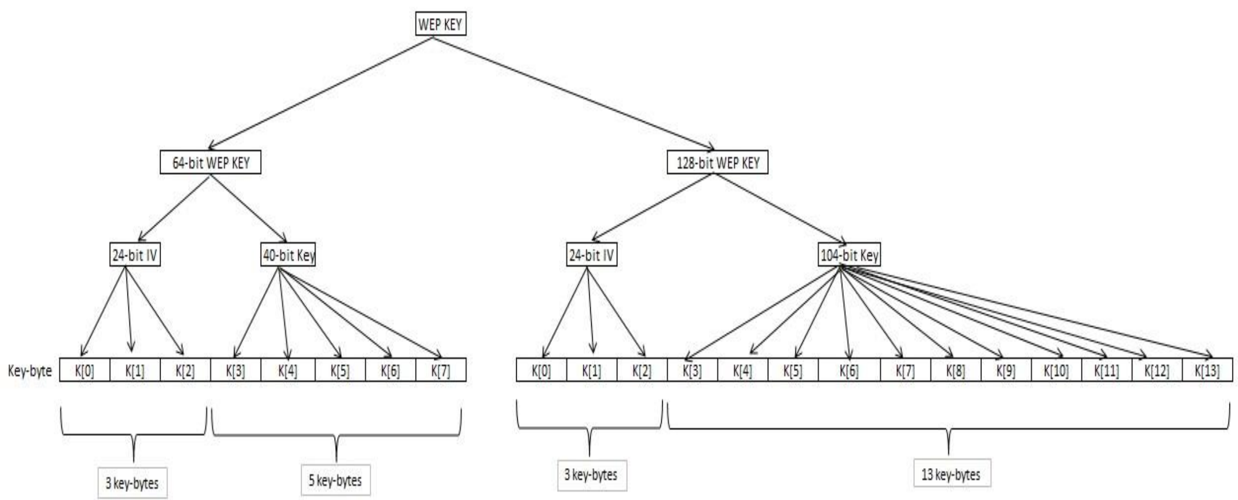


Figure 102: The 64-bit and 128-bit WEP key-bytes

During cracking of the WEP key, we must correctly guess the first true key-byte (K[3]) before we can obtain the remaining key bytes. This makes the attack statistical in nature as each weak IV gives about 5% chance of guessing the correct key-byte and 95% chance of guessing wrongly. However, by analyzing a large number of these weak IVs and the key bytes they reveal, we can expect to see a bias towards the true key bytes.

That is why an ARP packet was captured and replayed back into the network to generate more IVs containing these weak IVs. Each weak IV provides a statistical vote for each unknown key byte as shown in figure 103.

All the obtained weak IVs with their corresponding key-bytes are ranked based on their statistical votes from most probable key-byte to the least probable key-byte as shown in figure 103. The correct key-bytes (except for the last key-byte) are displayed in the first column of figure 103; The numbers next to the key-bytes are the votes for these key-bytes, The numbers right to these values are the alternative candidates for the key-bytes and their votes.

The correct key is found by using a few IVs and the key to generate the corresponding PRGA. If the generated PRGA matches the ones returned by the captured packets, the key is assumed to be correct with a very high probability. If not, then at least one of the decisions for one of the keybytes must have been incorrect. The attack now start looking for a decision for a key-byte that it suspects to be wrong. It could choose a decision where the difference in the number of votes between the most voted value for the key-byte and the second most voted value for the key-byte is minimal. The attack now assumes that the correct key-byte is the second most voted one and continue the computation of the PRGA with the substitute key-byte. This is repeated until the correct key has been found or a time limit has been exceeded.

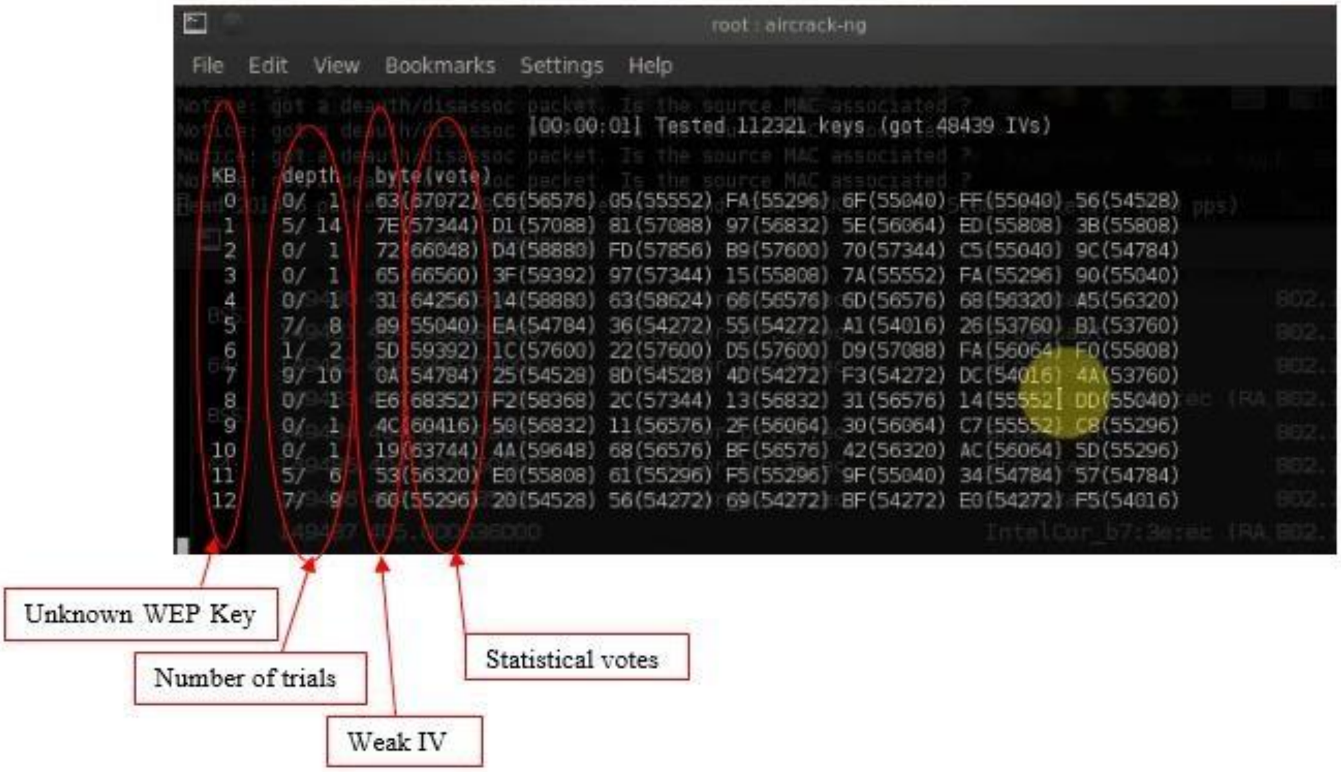


Figure 103: The “aircrack-ng” WEP cracking process

In our case, the WEP password was successfully cracked in less than 5 minutes after capturing 66,560 IVs and trying 541 possible keys as shown in figure 71.

The significance of this outcome is that WEP password cracking has about 99.9% success rate. This is due to the vulnerabilities in the use of ICV and presents of weak IVs which have a statistical correlation with the WEP password. So long as enough packets with weak IVs are obtained, the WEP key will be cracked. It does not depend on the size of the key, length of the password, or how complex the password is.

4.4 Analysis of the vulnerabilities in WPA/WPA-2 PSK

Message 1 of the EAPOL handshake which contains the ANounce and the Authenticator Mac Address were successfully obtained: This because message 1 of the EAPOL handshake is plaintext and can be eavesdropped by any adversary.

Message 2 of the EAPOL handshake which contains the SNounce, the Supplicant Mac Address, and the MIC were successfully obtained: This because message 2 of the EAPOL handshake is also plaintext and can be eavesdropped by any adversary.

The PTK which is a known derivative function of the ANounce, AP Mac Address, SNounce, Supplicant Mac Address, and the PMK was successfully computed:

PTK = Function (PMK, ANounce, SNounce, Authenticator Mac Address, Supplicant Mac Address).

From the above equation, the only unknown to an attacker is the PMK. However, recall too that the PMK is also another known derivative function of the Passphrase, SSID, SSIDLength, hashed 4096 times, to output a 256 bit key which is the PMK:

PMK = PBKDF2 (PassPhrase, ssid, ssidLength, 4096, 256).

By combining the above two equations, the PTK now becomes:

PTK = Function ([PBKDF2 (PassPhrase, ssid, ssidLength, 4096, 256)], ANounce, SNounce, Authenticator Mac Address, Supplicant Mac Address).

The only unknown now in the derivation of the PTK is the PassPhrase.

The PassPhrase was successfully found within the dictionary of the attacker: The attacker made guesses of the PassPhrase and computed the PTK for each guess. The attacker then computed an MIC per each computed PTK and compared with the captured MIC in Message 2 of the EAPOL Handshake.

There is a non-encrypted MIC in Message 2 of the EAPOL Handshake which was successfully used to ensure that the guessed PassPhrase was correct: The Attacker computed the MIC for each guessed PassPhrase's PTK and compared it with the MIC in Message 2. If there was a match, the attacker knew that it has ended up deriving the same PTK as the legitimate user. Hence his computation for the PMK is also correct and he has the correct PassPhrase to the network the WPA/ WPA2-PSK network.

The significance of this outcome is that if the password to a WPA/WPA2-PSK network can be found in a dictionary of an attacker, it will be successfully cracked. The dictionary file of the attacker is editable. The setback to this attack is that the dictionary file is case sensitive. If the password to the network is not captured with its case sensitive nature in the dictionary, it will not be cracked.

4.5 Analysis of the Success of cracking WPA/ WPA2-PSK Passwords

The attack was successful because the password to the WPA/ WPA2-PSK network was found within the dictionary of the attacker.

4.6 Analysis of the vulnerabilities in WPA/WPA2-EAP MD5 Security Protocol

All the EAP-MD5 Challenge and response messages were successfully obtained: All the messages including the EAP-Message ID, RADIUS challenge value, and the EAP MD5 Hash value are all in plaintext and were successfully captured.

The EAP MD5 Hash which is a function of the EAP-Message ID, RADIUS challenge value, and user Password was successfully computed:

EAP MD5 Hash value = (EAP-Message ID + User Password + RADIUS challenge value).

From the above equation, the only unknown in the derivation of the MD5 Hash is the PassPhrase.

The PassPhrase was successfully found within the dictionary of the attacker: The attacker made guesses of the PassPhrase and computed the EAP MD5 Hash value for each guess. The attacker then compared the computed the EAP MD5 Hash value with the captured EAP MD5 Hash value in the response message of the MD5 Handshake.

There is a non-encrypted EAP MD5 Hash value in the response message of the MD5 Handshake which was successfully used to ensure that the guessed PassPhrase was correct: The attacker compared the computed the EAP MD5 Hash value with the captured EAP MD5 Hash value in the response message of the MD5 Handshake. If there was a match, the attacker knew that it had ended up deriving the same MD5 Hash as the legitimate user. Hence his computation for the MD5 Hash is also correct and he has the correct PassPhrase to the WPA/ WPA2-EAP MD5 network.

The significance of this outcome is that if the password to a WPA/ WPA2-EAP MD5 network can be found in a dictionary of an attacker, it will be successfully cracked. The dictionary file of the attacker is editable. The setback to this attack is that the dictionary file is case sensitive. If the password to the network is not captured with its case sensitive nature in the dictionary, it will not be cracked.

4.7 Analysis of the Success of cracking WPA/ WPA2-EAP MD5 Passwords

The attack was successful because the password to the WPA/ WPA2-EAP MD5 network was found within the dictionary of the attacker.

4.8 Analysis of the vulnerabilities in all WPA/WPA2-EAP Authentication Methods that support the use of only Server-Side digital Certificates (EAP-TTLS & PEAPv0)

The client blindly accepted a fake digital certificate: Once the client accepted the fake digital certificate, a TLS tunnel was successfully created for the exchange of the MSCHAPv2 handshake between the attacker and the victim's machine.

All the MSCHAPv2 Challenge and response messages were successfully obtained: All the messages including the EAP-Message ID, RADIUS challenge value, and the MSCHAPv2 Hash value are all in plaintext and were successfully captured.

The MSCHAPv2 Hash which is a function of the EAP-Message ID, RADIUS challenge value, and user Password was successfully computed:

MS-CHAPv2 Hash value = (EAP-Message ID + User Password (from the user database) + RADIUS Challenge value)

From the above equation, the only unknown in the derivation of the MSCHAPv2 Hash is the PassPhrase.

The PassPhrase was successfully found within the dictionary of the attacker: The attacker made guesses of the PassPhrase and computed the MSCHAPv2 Hash value for each guess. The attacker then compared the computed MSCHAPv2 Hash value with the captured MSCHAPv2 Hash value in the response message of the MSCHAPv2 Handshake.

There is a non-encrypted MSCHAPv2 Hash value in the response message of the MSCHAPv2 Handshake which was successfully used to ensure that the guessed PassPhrase was correct: The attacker compared the computed the MSCHAPv2 Hash value with the captured MSCHAPv2 Hash value in the response message of the MSCHAPv2 Handshake. If there was a match, the attacker knew that it had ended up deriving the same MSCHAPv2 as the legitimate user. Hence his computation for the MSCHAPv2 Hash is also correct and he has the correct PassPhrase to the WPA/ WPA2-EAP TLS/PEAP network.

The significance of this outcome is that if the password to a WPA/ WPA2-EAP TLS/PEAP network can be found in a dictionary of an attacker, it will be successfully cracked. The dictionary file of the attacker is editable. The setback to this attack is that the dictionary file is case sensitive. If the password to the network is not captured with its case sensitive nature in the dictionary, it will not be cracked.

4.9 Analysis of the Success of cracking Passwords of WPA/WPA2EAP Authentication Methods that support the use of only ServerSide digital Certificates (EAP-TTLS & PEAPv0)

The attack was successful because the password to the WPA/ WPA2-EAP TLS/PEAP network was found within the dictionary of the attacker.

4.10 Analysis of the research survey

In all, 1,271 Access Points were surveyed. Out of this, 260 APs did not have any security credentials configured on them. This means that a hacker can easily join the network without requiring authentication. These Open Authentication APs represented 20.5% of the total APs surveyed.

104 APs had WEP security credentials configured on them. This means that these APs are vulnerable to the WEP attacks discussed in this thesis work. These WEP encrypted APs represented 8.1% of the total APs surveyed.

106 APs had WPA-PSK-TKIP security credentials configured on them. This means that these APs are vulnerable to WPA Passphrase or Dictionary attacks if the passphrases can be found in an attacker's dictionary. These WPA-PSK-TKIP encrypted APs represented 8.3% of the total APs surveyed.

55 APs had WPA-PSK-CCMP security credentials configured on them. This means that these APs are also vulnerable to WPA Passphrase or Dictionary if the passphrases can be found in an attacker's dictionary. These WPA-PSK-CCMP encrypted APs represented 4.3% of the total APs surveyed.

100 APs had WPA2-PSK-TKIP security credentials configured on them. This means that these APs are also vulnerable to WPA2 Passphrase or Dictionary attacks if the passphrases can be found in an attacker's dictionary. These WPA2-PSK-TKIP encrypted APs represented 7.9% of the total APs surveyed.

626 APs had WPA2-PSK-CCMP security credentials configured on them. This means that these APs are also vulnerable to WPA2 Passphrase or Dictionary if the passphrases can be found in an attacker's dictionary. These WPA2-PSK-CCMP encrypted APs represented 49.2% of the total APs surveyed.

30 APs had WPA2-EAP-CCMP security credentials configured on them. This means that these APs support EAP Authentication methods such as EAP-TLS, EAP-TTLS, PEAPv0 and inner authentication methods such

as MS-CHAPv2 or MD5 challenge and response messages. As discussed in section 4.4, these EAP methods are vulnerable to password if they do not provide mandatory support for both server-side and client-side certificates, and their passwords can be found in an attacker's dictionary. These WPA2-EAP-CCMP encrypted APs represented 2.4% of the total APs surveyed.

No APs were found to support WPA-EAP-TKIP, WPA-EAP-CCMP, and WPA2-EAP-TKIP security credentials.

Table 3 provides a summary of the results of the survey for the 1.271 Access Points discovered in the Greater Accra Region of Ghana

WLAN Security Credentials	Number of APs	Percentage of APs
No Security	260	20.5%
WEP	104	8.1%
WPA-PSK-TKIP	106	8.3%
WPA-PSK-CCMP	55	4.3%
WPA-EAP-TKIP	0	0%
WPA-EAP-CCMP	0	0%
WPA2-PSK-TKIP	100	7.9%
WPA2-PSK-CCMP	626	49.2%
WPA2-EAP-TKIP	0	0%
WPA2-EAP-CCMP	30	2.4%

Table 3: A summary of the results of the survey of 1,271 APs:

CHAPTER 5

FINDINGS, CONCLUSIONS, RECOMMENDATIONS AND AREAS FOR FUTURE RESEARCH

5.1 Overview

This chapter concludes this thesis work on "Vulnerability Analysis in Wireless Local Area Networks: A Survey of Some Wireless Access Points in Ghana". It provides the findings from the research, conclusions, recommendations, and areas for future research.

5.2 Findings of the Thesis work

1. WEP key cracking does not depend on the size of the key: It takes apparently the same time to crack a 64-bit and a 128-bit WEP key.
2. WEP key cracking depends on the number of weak IVs. It is suggested to gather between 60,000 to 70,000 IVs. Our attack was successful with 66,560 IVs.
3. It is faster to generate more weak IVs by capturing a gratuitous ARP packet, modifying it for the same host, and injecting it back into the network. This will generate more ARP request and response messages with IVs and keystream bytes.
4. WEP key can also be cracked offline. After gathering enough packets with weak IVs from the network, the data can be written to a file and bruteforced offline.
5. WPA/ WPA2-PSK Passwords can only be cracked if the password can be found in an attacker's dictionary.
6. The attacker's dictionary is editable.
7. The attacker's dictionary is case-sensitive.

8. WPA/ WPA2-EAP Inner Authentication methods such as MD5, MSCHAPv2, PAP, and etc are all plaintext messages.
9. All secured tunnels which carry WPA/ WPA2-EAP Inner Authentication methods that do not mandate the verification of both server-side and client-side digital certificates are vulnerable to accepting fake digital certificate.
10. WPA/ WPA2-EAP TLS could not be attacked because it mandates the verification of both server-side and client-side digital certificates. This was because the attacker had to first get a copy of the public key of the devices in the legitimate network. Because the attacker was not a member of the network, it could not prove its authenticity to the network devices. Hence the attack failed.

5.3 Conclusion of the Thesis work

From this thesis work, it was have proven that there are indeed vulnerabilities in Wireless Local Area Networks. The IEEE 802.11 WLAN security protocols (WEP, WPA/ WPA-2 PSK, and WPA/WPA-2 EAP) are vulnerable to various attacks using softwares that are freely available on the internet.

WEP suffered from the use of weak IVs, shorter IV space, and linear ICVs which led to the successful cracking of the password with 66,560 captured IVs in less than 5 minutes.

WPA/WPA2-PSK and WPA/WPA2-EAP suffered from the use of passwords which could be found in the attacker's dictionary which led to the successful cracking of the passwords.

WPA/WPA2-EAP TLS could not be cracked because it mandated the use of both server-side and client-side digital certificates.

5.4 Recommendations on minimizing attacks on WLAN infrastructure

1. WEP keys are static. It is recommended that users change the WEP keys as frequently as possible. This will minimize the risk of been cracked and used on the network.
2. WEP key cracking provides about 99.9% success rate if enough IVs (between 60,000 to 70,000 IVs) are gathered. These number of packets can be gathered within less than 5 minutes. It is recommended for users and administrators to stop using WEP completely.
3. WPA/ WPA2-PSK Passwords can only be cracked if the password can be found in an attacker's dictionary. It is recommended that users and administrators do not use default passwords that come with their Wifi devices.
4. The attacker's dictionary is editable. It is recommended that users do not use passwords that can be found on the internet.
5. The attacker's dictionary is case-sensitive. It can only crack the password if it is the same and case-sensitive as the one in the attacker's dictionary. It is recommended that users and administrators use alphanumeric passwords with a mixture of case-sensitive characters.
6. WPA/ WPA2-EAP MD5 does not support the use of secured tunnels to protect it from eavesdropping. It is fully plaintext. It is recommended for users and administrators to completely stop deploying WPA/WPA2-EAP MD5 on their networks.
7. WPA/ WPA2-EAP TTLS and PEAP mandate only server-side digital certificates. It is therefore vulnerable to accepting a fake server-side digital certificate. It is recommended that users and administrators scrutinize digital certificates very well before accepting them. To successfully validate certificates, users of Windows and other OS should ensure that the following features are always checked:
    a. "**Validate Server certificate**" should always be ticked as shown in figure 104.

b. “**Connect to these servers**” should always be ticked and select a list of trusted certificates you are very sure that they were issued by your legitimate RADIUS Server as shown in figure 104.

c. “**Do not prompt user to authorize new servers or trusted certification authorities**” options should also be ticked as shown in figure 102 so that if another certificate is issued by a server outside the list of trusted servers, the legitimate client will reject it.

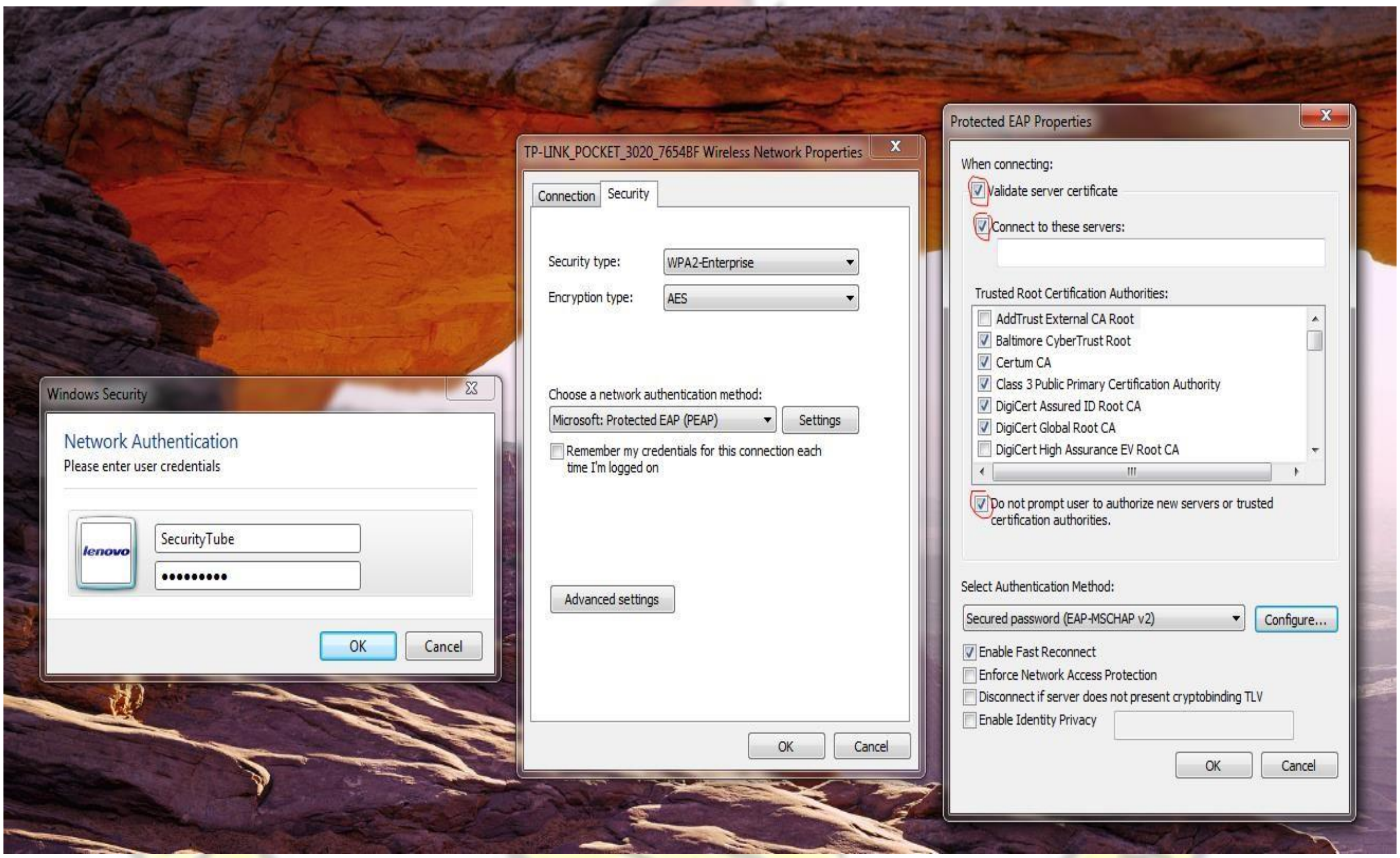


Figure 104: The client validating digital certificates to prevent accepting fake digital certificates

8. If resources are available, it is recommended that enterprise users use EAP-TLS authentication method. This mandates the use of both client-side and server-side certificates. It is not vulnerable to any attack as at the time of writing this thesis work.

9. Organizations should create and enforce wireless network security policies that address all the known vulnerabilities. Such policies should include which users are allowed to use the WLANs and what level of information is allowed to be transmitted over the WLANs.
10. Security assessments or audits are essential for checking the security posture of an organizations' WLAN infrastructure. It is important for organizations to perform regular audits of their WLANs to identify rogue APs and unauthorized access.
11. Organizations could also outsource the regular network audit to a third-party who have the tools and the technical expertise to do a more detailed penetration testing and put in measures to minimize a possible attack on the network.

5.5 Areas for future research

This thesis work have discovered some vulnerabilities in the IEEE 802.11 security protocols of Wireless Local Area Networks. Experiments were carried out to prove these vulnerabilities and to successfully crack the passwords.

However, there are possibilities of other vulnerabilities, attacks, and defense mechanisms that can be exploited in 802.11 WLANs. The following research areas are provided as a guide to be exploited:

1. There are possibilities of other vulnerabilities and attacks that can be exploited in 802.11 WLANs. It will be good for one to conduct research into these areas.
2. Due to time constraints, this thesis work could not explore into patching the vulnerabilities discovered in the IEEE 802.11 Security protocols. It is recommended for one to conduct research into patching these flaws.

3. The weakness in WEP is the way it implements RC4 algorithm which reveals information about the WEP key. It will be good for one to conduct research into how the RC4 Algorithm reveals information about the WEP key and possibly provide a patch.
4. In this thesis work, WEP was broken with the presence of both the access point and the client. It should be possible to create a rogue AP and lure the client to authenticate with it and break the WEP key with just the client. It will be good for one to research into this area.
5. WPA/ WPA2-PSK can be hacked using only a dictionary attack for now. This makes the attack less certain if the password is not in the dictionary. To completely consider WPA/WPA2-PSK flawed, the attack must be statistical in nature. If the password is hashed 4096 times to output the 256-bit PMK, it should be possible for one to research into reverse engineering the PMK to reveal the password and possibly model the attack statistically.
6. For now, WPA/WPA2-EAP TLS is the most secured WLAN because it mandates the validation of both server-side and client-side digital certificates. To be able to break this, one must be able to capture a public key from a device in the legitimate network. The fake server can use this to prove its authenticity to the client which will lead to a successful creation of fake secured tunnel to allow the passage of the inner authentication protocols for a bruteforce attack. It will be good for one to research into this area.
7. There are commercially available Network Intrusion Detection Systems. However less literature are available on the success and failures of their use. It will be great if one researches into the pros and cons of Network Intrusion Detection Systems (NIDS).

REFERENCES

1. Abdul Qudoos Memon, Ali Hassan Raza and Sadia Iqbal, "*WLAN Security*", April 2010.
2. Aboba B & Simon D. "*PPP EAP TLS Authentication Protocol*". RFC 2716. October 1999.
3. Aboba B. "*Pros and Cons of Upper Layer Network Access*". IEEE Documents 802.11. November 2000.
4. Abrahamsson Charlotte & Wessman Mattias, "*WLAN Security: IEEE 802.11b or Bluetooth-Which Standards provides best security methods for companies?*". May 2004.
5. Adelstein, F., Alla, P., Joyce, R. & Richard III, G.G, *"Physically locating wireless intruders*". International Conference on Information Technology: Coding and Computing. May 2004.
6. Akin Devin. "*802.11i Authentication and Key Management (AKM)*". Whitepaper. Certified Wireless Network Professional. Planet3 Wireless Inc. May 2005.
7. Arbaugh W.A, Shankar N, Wang J, & Zhang K. "*Your 802.11 Network has No Clothes"*. Suntec City, Singapore. http://citeseer.nj.nec.com/566520.html .March 2001.
8. Arbaugh W.A. "*An Inductive chosen Plaintext Attack against WEP/WEP2*". http://www.cs.umd.edu/~waa/attack/v3dcmnt.htm . June 2001.
9. Beck A. "*NetScape's Export SSL broken by 120 Workstations and one student*". HPCoorre. August 1995.
10. Biham E & Carmeli Y. "*Efficient Reconstruction of RC4 Keys from Internal States*". In Kaisa Nyberg, editor, FSE, volume 5086 of Lecture Notes in Computer Science, pages 270-288. Springer. November 2008.

11. Bittau Andrea, Handley Mark, & Lackey Joshua. "*The Final Nail in WEP's Coffin*". In IEEE Symposium on Security and Privacy. Pages 386-400. IEEE Computer Society. July 2006.

12. Bittau Andrea. "*Additional Weak IV Classes for the FMS Attack*". http://www.cs.ucl.ac.uk/staff/a.bittau/sorwep.txt . June 2003.

13. Bittau Andrea. "*The Fragmentation Attack in Practice*". September 2005.

14. Borisov N, Goldberg I, & Wagner D. "*Intercepting Mobile Communications: The Insecurity of 802.11*". In Proc. ACM Mobicom, Rome, Italy. July 2001.

15. Bradra M & Hecker A. "*Security in WLAN*". IGI Global Handbook of Research on Wireless Security. Vol 1. September 2008.

16. Bryan L. Bradford, "*Wireless Security within Hastily Formed Networks*", September 2006.

17. Carter B & Shumway R, "*Wireless Security End to End*". Indianapolis, Indiana: Wiley Publishing. August 2002.

18. Chaabouni Rafik. "*Break WEP Faster with Statistical Analysis*". Technical Report. EPFL, LASEC. June 2006.

19. Chakraborty S. "*Understanding EAP-MD5 Authentication with RADIUS*". ISmart International Limited. June 2005.

20. Chandra Praphul, Bensky Alan, Bradley Tony, Hurley Chris, Rackley Steve, Rittinghouse John, Ransome F.James, Stapko Timothy, Stefanek L.George, Thornton Frank, & Wilson John. "*Wireless Security, Know it all*". ISBN 978-1-85617-529-6. Elsevier Inc. May 2009.

21. Chandra Shekar Gambiraoput, "*Security Threats and Intrusion Detection Models in*

WLAN", January 2010.

22. Changhua He, "*Analysis of Security Protocols for Wireless Networks*", December 2005.

23. Chen Hsiao-Hwa & Guizani M. "*Next Generation Wireless Systems and Networks*". John Wiley & Sons Ltd. The Atrium, Southern Gate, Chichester, West Sussex P019 8SQ. England, March 2006.

24. Chen, Cheng I, Jiang, Chia M, Liu, & Wen Y. "*Wireless LAN Security and IEEE 802.11i*". IEEE Wireless Communications. February 2005.

25. Ciampa M, "*Guide to Designing and Implementing Wireless LANs*". Course Technology. October 2001.

26. Davis D. "Wireless Networking CWNA Study Package. Managing Wireless Networks for the Blue Crab Food Company". TrainSignal Incorporated. www.triansignal.com . April 2005.

27. Dr.Norman F. Schneidewind, "IS3502 Computer Networks: WAN/ LAN Wireless Networks", couse notes, Naval Postgraduate School, 2002.

28. Edney J & Arbaugh W.A. "*Real 802.11 Security: Wi-Fi Protected Access and 802.11i*". Addison-Wesley. Bouston. May 2004.

29. Fluhrer S.R, Mantin I, & Shamir A. "*Weaknesses in the Key Scheduling Algorithm of RC4*". In Serge Vaudenay and Amr M. Youssef, editors. Selected Areas in Cryptography 2001.Vol 2259 of Lecture Notes in Computer Sciences, pages 1-24. Springer. August 2001.

30. Foust R. "*Identifying and Tracking Unauthorised 802.11 Cards and Access Points. A Practical Approach*". August 2002.

31. Frankel Sheila, Eydt Bernard, Owens Les, & Karen Scarfone. "*Establishing Wireless Robust Security Networks: A Guide to IEEE 802.11i*". Recommendations of the National Institute of Standards and Technology, NIST Special Publications, February 2007.

32. Gambiraopet C S. "*Security Threats and Intrusion Detection Models in WLAN*". October 2009.

33. Garg V. "*Wireless Communications and Networking*". June 2001.

34. Gast S.M, "*802.11 Wireless Networks: The Definitive Guide*". O'Reilly & Associates, ISBN: 0596001835. April 2002.

35. Goldsmith C. "*Wireless Local Area Networking for Device Monitoring*". June 2004.

36. Golic. "*Linear Statistical Weakness of Alleged RC4 Keystream Generator*". In EUROCRYPT: Advances in Cryptography: Proceedings of EUROCRYPT. June 1997.

37. Goutam Paul, Siddheshwar Rathi, and Subhamoy Maitra. "*On Non-Negligible Bias of the First Output Bytes of RC4 towards the First Three Bytes of the Secret Key*". In WCC'07-International Workshop on Coding and Cryptography. Pages 285-294. March 2007.

38. Halvorsen F.M & Haugen O. "*Cryptanalysis of IEEE 802.11i TKIP*". Master of Science in Communication Technology. Norwegian University of Science and Technology. June 2009.

39. Hanifa Abdulallah, "*A Risk Analysis and Risk Management Methodology for Mitigating Wireless Local Area Networks (WLANs) Intrusion Security Risks*", April 2006.

40. Hasan-Al-Banna, Riaz Mahmood Rajib, "*Wireless LAN 802.11 Security*", January 2008.

41. Haykin Simon. "*Modern Wireless Communication*". ISBN 0-13-124697-6. Prentice Hall. August 2005.

42. Hoe Keng. "*Security Guidelines for Wireless LAN Implementation*". SANS Institute InfoSec Reading Room. GIAC Security Essentials Certification (GSEC). August 2003.

43. Hutton D. "*Practical Exploitation of RC4 Weakness in WEP Environments*". Presented at Hiver in 2002. June 2002.

44. IEEE, IEEE 802.11 Standards documents. http://standards.ieee.ogr/wireless/ . January 2004.

45. J. Duntemann, "*Wardriving FAQ*", April 2003.

46. Joon S.Park and Derrick Dicoi, "*WLAN Security: Current and Future*", Syracuse University. Published by the IEEE Computer Society, IEEE Internet Computing. September 2003.

47. Kazukuni Kobara & Hideki Imai. "*IVs to Skip for Immunizing WEP against FMS Attack*". IEICE Trans. Fundamentals. Volume E91-B. No.1. January 2008.

48. Keran Tan, "*Large-Scale Wireless Local Area Network Measurement and Privacy Analysis*", August 2011.

49. Khan J & Khwaja A. "*Building Secure Wireless Networks with 802.11*". Canada. Wiley Publishing, Indianapolis, Indiana. June 2003.

50. Kizza J.M, "*A Guide to Computer Network Security*", June 2008.

51. Klein Andreas. "*Attacks on the RC4 Stream Cipher*". Submitted to Designs, Codes and Cryptography. August 2006.

52. Lasse Seppanen, "*Wireless Local Area Network (WLAN) IEEE 802.11*", whitepaper, 2002, http://trade.hank.fil-/seppane/courses/wlan/doc/material.pdf , February 2002.

53. Lewis B.D & Davis P.T. "*Wireless Networks for Dummies*". Indianapolis, Indiana:Wiley Publishing. June 2004.

54. Madjid Nakhjiri & Mahsa Nakhjiri. "*AAA and Network Security for Mobile Access: RADIUS, DIAMETER, EAP, PKI and IP Mobility*". John Wiley & Sons Ltd. The Atrium, Southern Gate, Chichester, West Sussex P0198SQ, England. ISBN-13 978-0470-01194-2. January 2005.

55. Mantin Itsik. "*A Practical Attack on the Fixed RC4 in the WEP Mode*". In Bimal K. Roy, editor, ASIACRYPT. Volume 3788 of Lecture Notes in Computer Science. Pages 395-411. Springer. July 2005.

56. Mark J.W & Zhuang W. "*Wireless Communications and Networking*". Upper Saddle River, NJ. Prentice Hall, July 2003.

57. Masica Ken. "*Recommended Practices Guide: Securing WLANs using 802.11i*". Vulnerability & Risk Assessment Program (VRAP). Lawrence Livermore National Laboratory (LLNL). October 2006.

58. Matsui M. "*The First Experimental Cryptanalysis of the Data Encryption Standard*". Proceedings of the 14th Annual International Cryptology Conference on Advances in Cryptology. Pages 1-11, February 1994.

59. Menezes A, Oorschot van P, & Vanstone S. "*Handbook of Applied Cryptography*". CRC Press Inc. July 1997.

60. Mironov Ilya. *"(Not so) Random Shuffles of RC4*". In Moti Yung, editor, CRYPTO. Volume 2442 of Lecture Notes in Computer Sciences. Pages 304-319. Springer. August 2002.

61. Nedeltchev Plamen, "*Wireless Local Area Networks and the 802.11 Standard*".

Whitepaper. http://www.cisco.com/warp/public/84/packet/jul01/pdfs/whitepaper.pdf . March 2001.

62. Nwabude Arinze Sunday, "*Wireless Local Area Network:Security Risk Assessment and Countermeasures*", August 2008.

63. P. Shipley, "*Open WLANs: The early result of Wardriving*", 2001.

64. Pahlavan K, & Krishnamurthy P. "*Principle of Wireless Networks: A Unified Approach*". Upper Saddle River, NJ. Prentics Hall, May 2002.

65. Peikari C, & Forgie S. "*Maximum Wireless Security*". Macmillan Computer Pub, ISBN 0672324881. December 2002.

66. Rackley Steve, "*Wireless Networking Technology. From Principles to Successful Implementation*". May 2007.

67. Regan K. "*Wireless LAN Security: Things You Should Know about WLAN Security*". Network Security. March 2003.

68. Russell Dean Vines, "*Wireless Security Essentials*", Wiley Publishing, Inc. 2002.

69. Scarfone K & Souppaya M. "*Guidelines for Securing Wireless Local Area Networks (WLANs)*". Recommendations of the National Institute of Standards and Technology. Computer Security Division. Gaithersburg, MD 20899-8930. February 2012.

70. Scarfone K, Dicoi D, Sexton M, & Tibbs C. "*Guide to Securing Legacy IEEE 802.11 Wireless Networks*". Recommendations of the National Institute of Standards and Technology. Computer Security Division. Gaithersburg, MD 20899-8930. July 2008.

71. Shafi M, Ogose S, & Hattori T (editors), "*Wireless Communication in the 21st Century*". Wiley-Interscience, February 2002.

72. Sithirasenan E, Muthukkumarasamy V, & Powell D. "*IEEE 802.11i WLAN Security*

Protocol: A Software Engineer's Model". School of Information and Communication Technology. Griffith University. Queensland, Australia. August 2005.

73. Sithirasenan E. "*A Preliminary Analysis of the IEEE 802.11i WLAN Protocol*". Masters Thesis, Griffith University. Australia. October 2004.

74. Souradyuti P & Prencel B. "*A New Weakness in the RC4 Keystream Generator and an Approach to Improve the Security of the Cipher*". In Bimal K. Roy & Willi Meter, editors, FSE. Volume 3017 of Lecture Notes in Computer Science. Pages 245-259. Springer. July 2004.

75. Stallings W. '*Cryptography and Network Security: Principles and Practice*".Fifth Edition. Prentice Hall. Lake Street, Upper Saddle River, NY 07458. ISBN 13: 978-013-60 9704-4. August 2006.

76. Stallings W. "*Wireless Communications and Networks*". Prentice Hall USA. August 2002.

77. Stubblefield A, Ioannidis J, & Rubin D.A. "*A Key Recovery Attack on the 802.11b Wired Equivalent Privacy Protocol (WEP)*". ACM Transactions on Information and System Security. Pages 319-322. May 2004.

78. Stubblefield A, Ioannidis J, & Rubin D.A. "*Using the Fluhrer, Mantin, and Shamir Attack to Break WEP)*". May 2002.

79. Tanzella F."*Wireless LAN Intrusion Detection & Protection*". http://www.airdefense.net. August 2003.

80. Tews E & Beck M. "*Practical Attacks against WEP and WPA*". In David A. Basin, Srdjan Capkun, and Wenke Lee, editors, WISEC. Pages 79-86, ACM. October 2009.

81. Tews E, Ralf-Philipp Weinman, & Pyshkin A. "*Breaking 104 bits WEP in Less Than 60 seconds*". In Sehum Kim, Moti Yung, and Hyung-Woo Lee, editors, WISA, volume 4867 of Lecture Notes in Computer Science, pages 188-202, Springer. July 2007.

82. Tews Erik. "*Attacks on the WEP Protocol*". Diploma Thesis Fachgebiet Theoretische Informatik. December 2007.

83. Thoetsak Jaiaree, "*The Security Aspects of Wireless Local Area Network (WLAN)*", September 2003.

84. Tom Karygiannis & Les Owens, "*Wireless Network Security 802.11, Bluetooth, and Handheld Devices 802.11*". National Institute of Standards and Technology, Administration, U.S. Department of Commerce, NIST Special Publication, November 2002.

85. Treschow M. "*Security in Wireless Local Area Networks. Advice to users for Improved Security*". June 2007.

86. Vivek Ramachandran. "*BackTrack 5 Wireless Penetration Testing. Beginners Guide*". ISBN 978-1-849515-58-0. September 2011.

87. Vyneke E & Paggen C."*LAN Switch Security: What Hackers know about your Switches-A practical guide to hardening Layer 2 devices and stopping campus network attacks*". Cisco Press. 800 East 96th Street, Indianapolis, IN46240.USA. ISBN 978-1-58705-256-9. August 2007.

88. Wagner David. "*Weak Keys in RC4 (sci.crypt)*". http://www.cs.berkeley.edu/-daw/myposts/my-rc4-weak-keys . September 1995.

89. Walker Jesse. "*802.11 Security Series. Part II. The Temporal Key Integrity Protocol (TKIP)*". Network Security Architect. Platform Networking Group. Intel Corperation.

June 2005.

90. Whalen Sean. "*Analysis of WEP and RC4 Algorithms*". Revision 1. http://www.chocobospore.org . March 2002.

91. Winget C.N, Moore T, Stanley D, & Walker J. "*IEEE 802.11i Overview*". September 2007.

Appendix A

Appendix A provides the first 32 ASCII character conversions into decimal, octal, and hexadecimal notations as discussed in section 2.2

Control Codes : ASCII Characters 0 - 31

The following table lists and describes the first 32 ASCII characters, often referred to as control codes. The columns show the decimal and hexadecimal ASCII values for each code along with their abbreviated and full names. Descriptions are given to those most in use today.

Decimal	Octal	Hexadecimal	Code	Description
000	000	00	NUL	Null
001	001	01	SOH	Start Of Heading
002	002	02	STX	Start of TeXt
003	003	03	ETX	End of TeXt
004	004	04	EOT	End Of Transmission
005	005	05	ENQ	ENQuiry
006	006	06	ACK	ACKnowledge

Decimal	Octal	Hexadecimal	Code	Description
007	007	07	BEL	BELL. Caused teletype machines to ring a bell. Causes a beep in many common terminals and terminal emulation programs.
008	010	08	BS	BackSpace. Moves the cursor move backwards (left) one space.
009	011	09	HT	Horizontal Tab. Moves the cursor right to the next tab stop. The spacing of tab stops is dependent on the output device, but is often either 8 or 10 characters wide.
010	012	0A	LF	Line Feed. Moves the cursor to a new line. On Unix systems, moves to a new line AND all the way to the left.
011	013	0B	VT	Vertical Tab
012	014	0C	FF	Form Feed. Advances paper to the top of the next page (if the output device is a printer).
013	015	0D	CR	Carriage Return. Moves the cursor all the way to the left, but does not advance to the next line.
014	016	0E	SO	Shift Out
015	017	0F	SI	Shift In
016	020	10	DLE	Data Link Escape
017	021	11	DC1	Device Control 1
018	022	12	DC2	Device Control 2
019	023	13	DC3	Device Control 3
020	024	14	DC4	Device Control 4
021	025	15	NAK	Negative AcKnowledge
022	026	16	SYN	SYNchronous idle
023	027	17	ETB	End of Transmission Block
024	030	18	CAN	CANcel
025	031	19	EM	End of Medium
026	032	1A	SUB	SUBstitute
027	033	1B	ESC	ESCape
028	034	1C	FS	File Separator

Appendix B

This appendix provides the instructional guide to setup the laboratory to test the various vulnerabilities that were discovered in the security protocols of WLANs during this thesis work as discussed in Chapters 3 and 4.

Laboratory setup requirements

1. Two laptops with internal Wi-Fi cards:
    a. One of the laptops will be used as the victim.
    b. The second laptop will be used as the attacker's laptop.
2. One Alfa network AWUS036NH wireless long-range usb card as shown in figure 105:
    a. Already integrated into BackTrack 5 software.
    b. Allows for packet sniffing, modification, and injection.
    c. Maximum output power is 5dBi.
    d. We will use this in all our experiments.
    e. It cost about $29 on http://www.amazon.com/s/ref=nb_sb_noss_2?url=searchalias%3Daps&field-keywords=alfa+network+AWUS036NH .

Figure 105: The Alfa network AWUS036NH wireless long-range USB card

3. An Access Point or Smartphone that supports the following security protocols:
    a. WEP
    b. WPA/ WPA2-PSK
    c. WPA/ WPA2-EAP

4. BackTrack 5 Software as shown in figure 106 which is a network Penetration Testing software as shown in figure :

   a. Can be downloaded freely from https://www.kali.org/downloads/

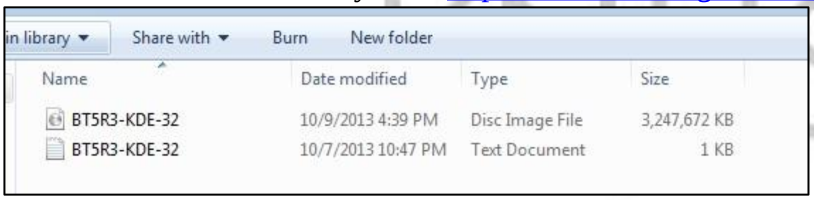


Figure 106: The freely downloaded BackTrack 5 Penetration Testing software

The Laboratory Setup Diagram

The laboratory setup includes a victim machine (supplicant), an Access Point, an Authentication server, a hacker machine (running BackTrack 5) and an Alfa AWUS036NH wireless card as shown in figure 107.

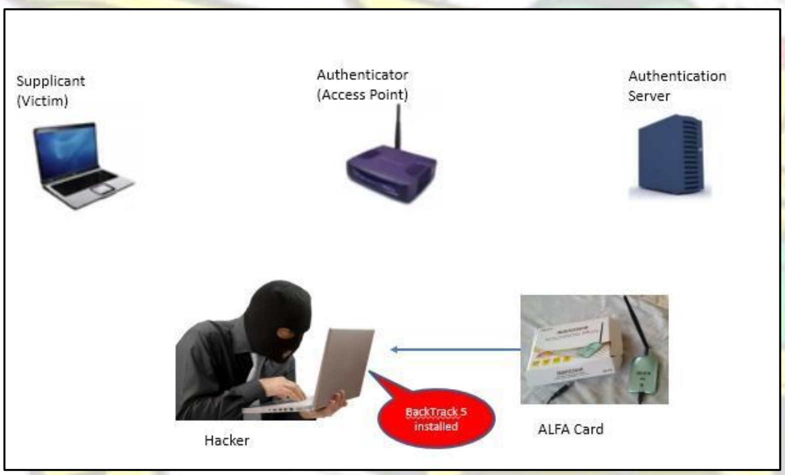


Figure 107: The laboratory setup diagram

Connecting the Alfa Card to the Attacker's Laptop

1. Connect the ALFA AWUS036NH card to the usb port of the attacker's laptop.

2. Download the ALFA card driver freely from http://www.alfa.com.tw/download_show.php?combo_0=&combo_1=&combo_2=&keyword=awus036nh&verify=523W85&x=11&y=7 and install on the attacker's laptop. Figure 108 shows the downloaded ALFA wireless card driver.

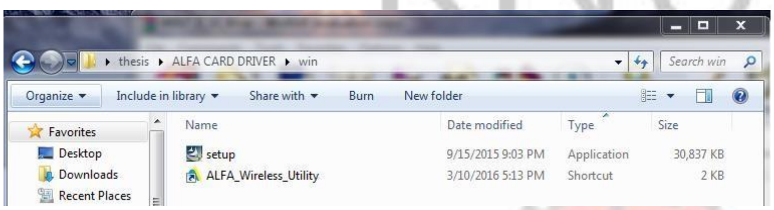


Figure 108: ALFA wireless card driver

Installing VirtualBox on the Attacker's Machine

1. Download and install VirtualBox on the attacker's laptop freely from https://www.virtualbox.org/wiki/Download_Old_Builds_4_1 . Figure 109 shows the virtualbox installed on the attacker's laptop.

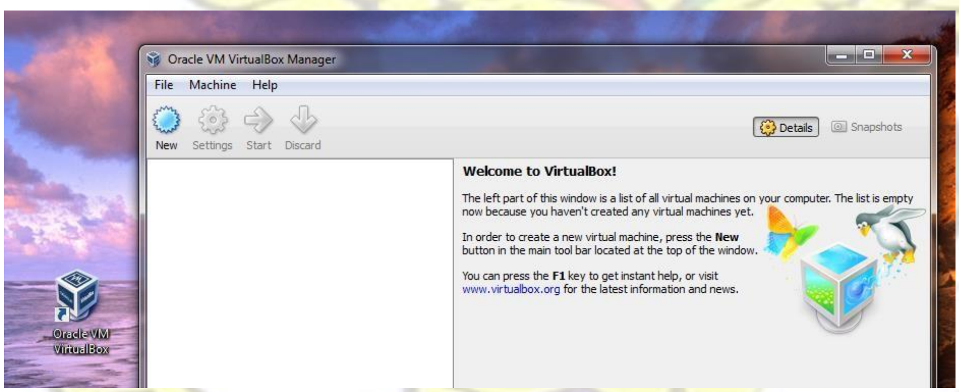


Figure 109: virtualbox installed on the attacker's laptop

Installing BackTrack5 in the VirtualBox of the Attacker's machine

1. Click on new and follow the wizard to load the Backtrack 5 software as shown in figures 110 and 111.

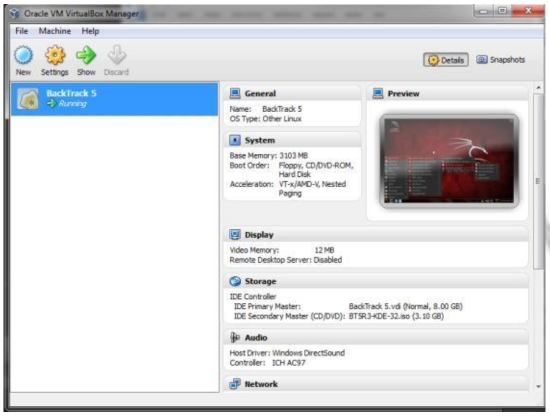


Figure 110: BackTrack 5 software loaded into VirtualBox

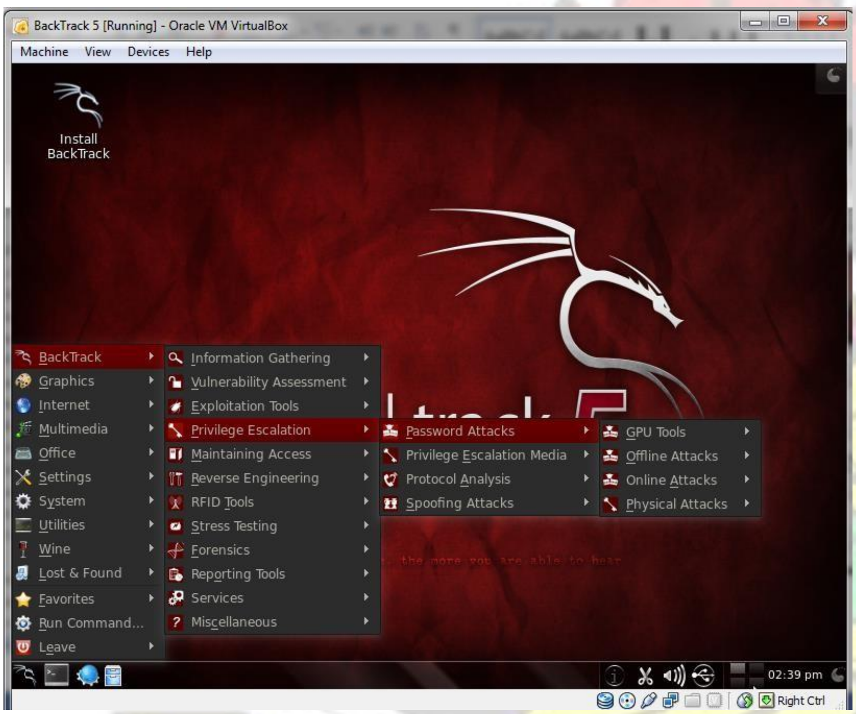


Figure 111: Console interface of BackTrack 5 Penetration Testing Software

Putting the Alfa card into Monitoring mode within BackTrack5

1. Connect the Alfa card to the attacker's laptop on which BackTrack5 is already running.

2. Run the commands “ifconfig, ifconfig wlan0 up, airmon-ng, airmon-ng start wlan0, and airmon-ng one after the other to put the alfa card into monitoring mode as shown in figure 112.

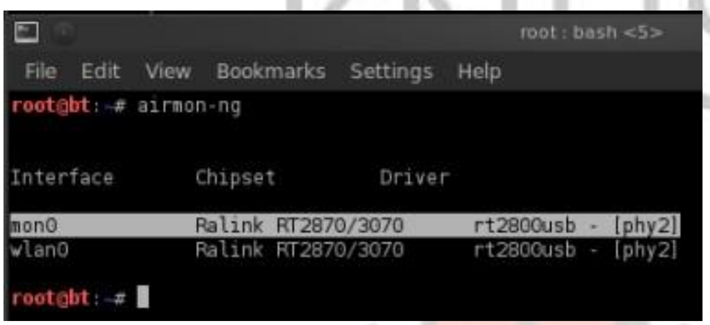


Figure 112: The ALFA card put into monitoring mode in BackTrack5

Setting up FreeRadius WPE Server within BackTrack5 within the Attacker’s Laptop to server as a RADIUS Authentication Server

For experimental purposes, it will be expensive to purchase an Authentication Server. BackTrack 5 comes with a free RADIUS-WPE Server software that mimics a real network Authentication Server.

FreeRadius server was set up on BackTrack as follows:

1. Configure network adapter 1 in the VM as a bridge to the NIC of the machine running BackTrack5. This will be configured as eth0 to connect to the Access Point as shown in figure 113.

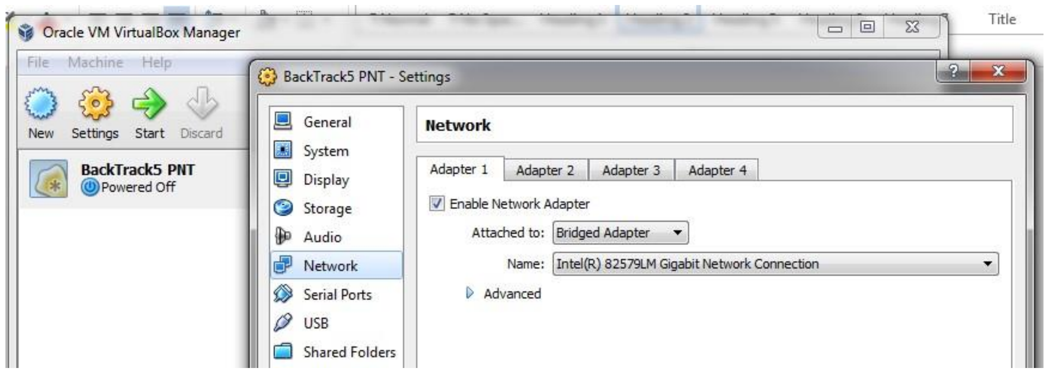


Figure 113: Bridging the VM to the NIC of the attacker's laptop

2. Configure network adapter 2 in the VM as a NAT to the WNIC of the machine running BackTrack5. This will be configured as eth1 to connect to the internet. Connect the WNIC to the internet as shown in figure 114.

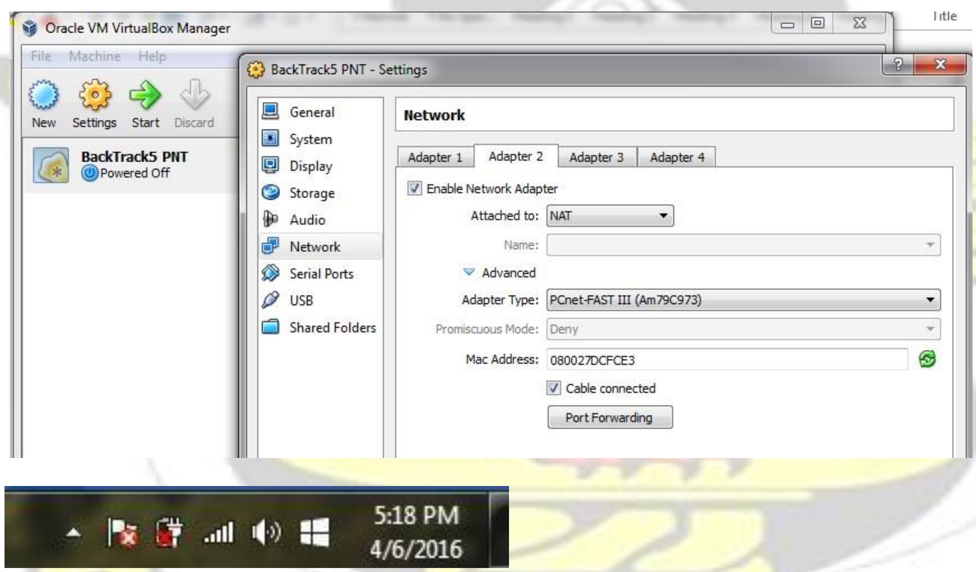


Figure 114: NAT the VM to the WNIC of the attacker's laptop

3. Use the command "dhclient3 eth1" to request IP from the internet through DHCP as shown in figure 115.

Figure 115: The results of the DHCP request from the internet

4. Ping 8.8.8.8 to check that you have connectivity to the internet as shown in figure 116.

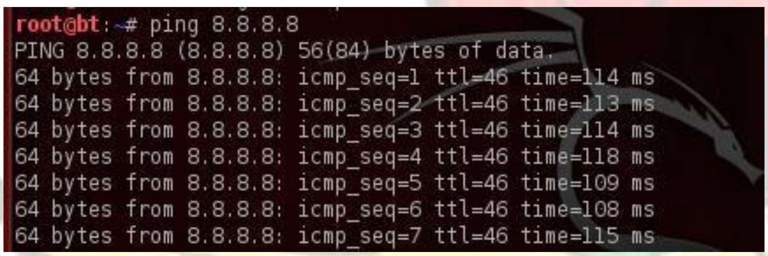


Figure 116: The results of a ping test to the internet

5. Download and save the openssl file from http://www.openssl.org/source/ as shown in figure 117.

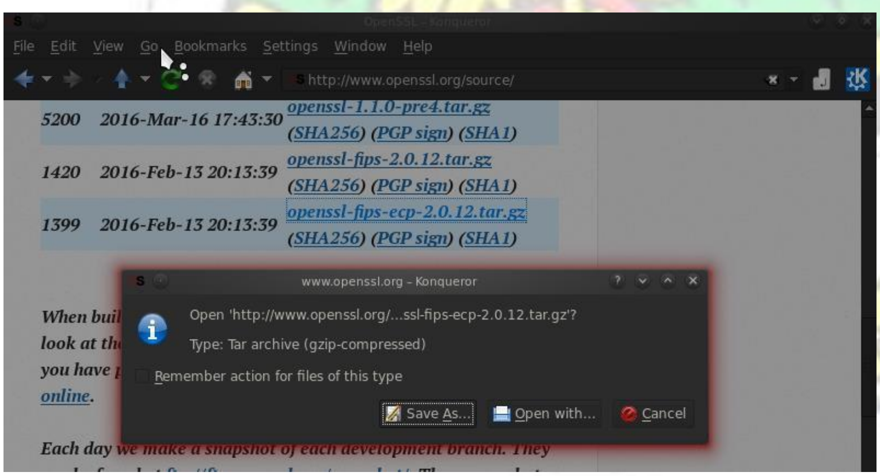


Figure 117: The procedure to download the openssl file

6. Save the openssl file to the desktop as shown in figure 119.

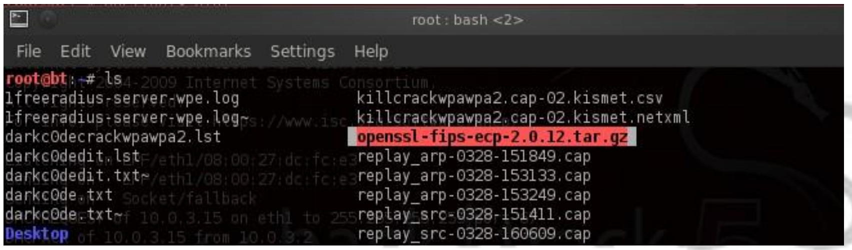


Figure 118: The location of the openssl file on the desktop

7. Use the command "tar -xvf" to extract the openssl file as shown in figure 120.

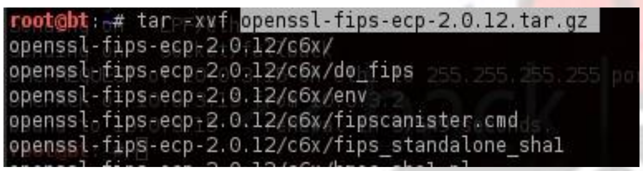


Figure 119: The command to extract the openssl file

8. Open the extracted openssl file as shown in figure 120.

Figure 120: The contents of the openssl file

9. Run the command "./config" under the openssl-fips-ecp-2.0.12 folder as shown in figure 121.

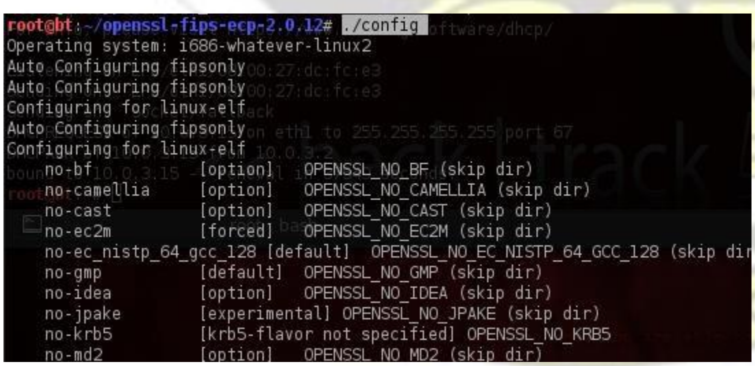


Figure 121: The results of the "/config" command

10. Run the next command “make” as shown in figure 122.

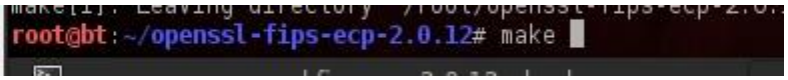


Figure 122: The results of the “make” command

11. Run the next command “make install” as shown in figure 123.

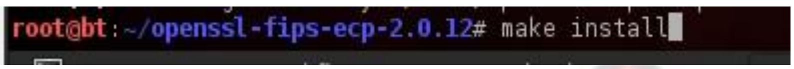


Figure 123: The results of the “make install” command

12. Download freeradius-server-2.2.9 from http://freeradius.org/download.html as shown in figure 124.

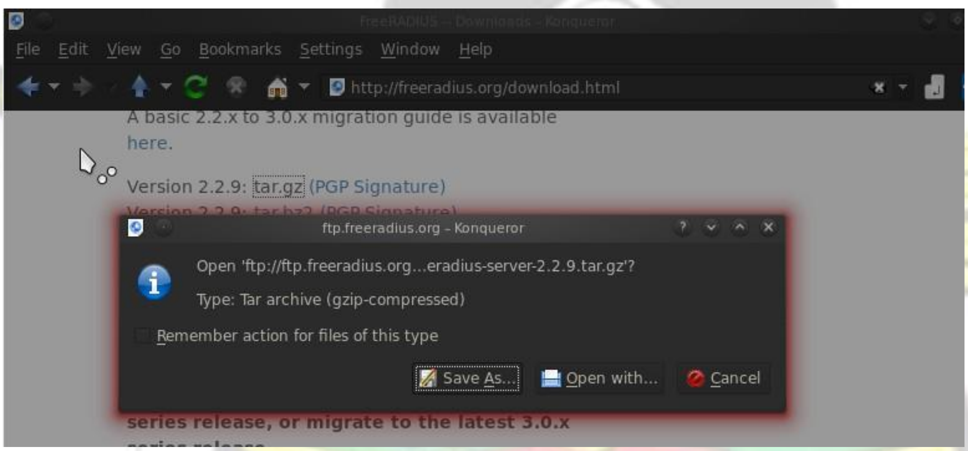


Figure 124: The procedure to download the freeradius-wpe server.

13. Save it to the desktop as shown in figure 125.

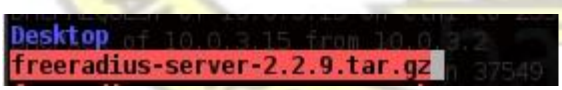


Figure 125: The location of the freeradius-wpe server on the desktop

14. Use the command “tar -xvf” to extract the freeradius-server file as shown in figure 126.

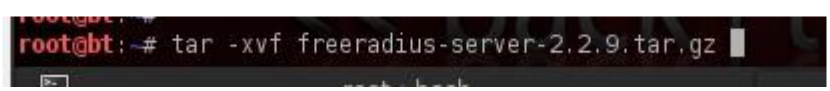


Figure 126: The results of the “tar -xvf” command

15. Open the extracted freeradius-server file as shown in figure 127.

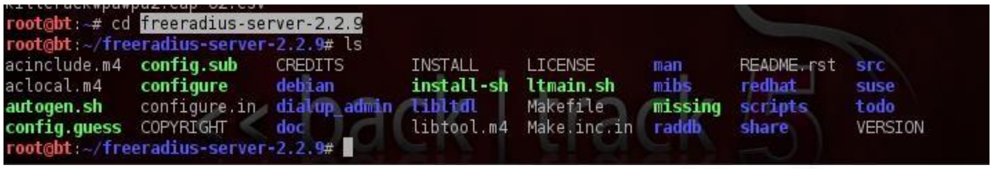


Figure 127: The contents of the freeradius-wpe server file

16. Run the command “./configure” under the freeradius-server-2.2.9 folder as shown in figure 128.

Figure 128: The results of the “./configure” command

17. Run the next command “make” as shown in figure 129.

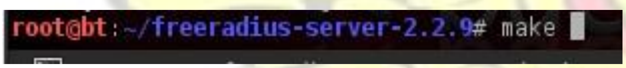


Figure 129: The “make” command

18. Run the next command “make install” as shown in figure 130.

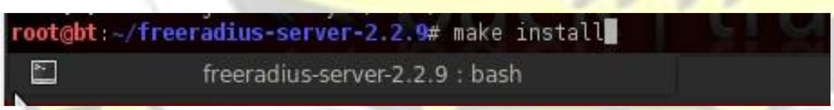


Figure 130: The “make install” command

19. Run the command “ /sbin/ldconfig -v" from root.

20. Configure a digital certificate that will be issued by the freeradius-wpe. Run the command to “./bootstrap” to use the default bootstrap certificates as shown in figure 134.

Figure 131: The freeradius digital certificate called bootstrap

21. Make a copy of the digital certificate and copy to usr/local/etc/raddb/certs/ as shown in figure 132.

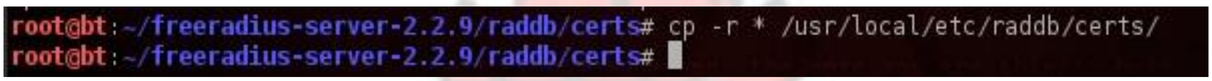


Figure 132: The command to make a copy of the digital certifate

22. Use the command "dhclient3 eth0" to request IP from the AP through DHCP as shown in figure 133.

Figure 133: The results of the DHCP request from the Access Point

23. Configure the Access Point to support WPA/ WPA-2 Enterprise. Set the Radius server IP to the IP obtained through DHCP in step 20. Configure the Radius Password which will be used to authenticate the AP to the Radius server as shown in figure 134.

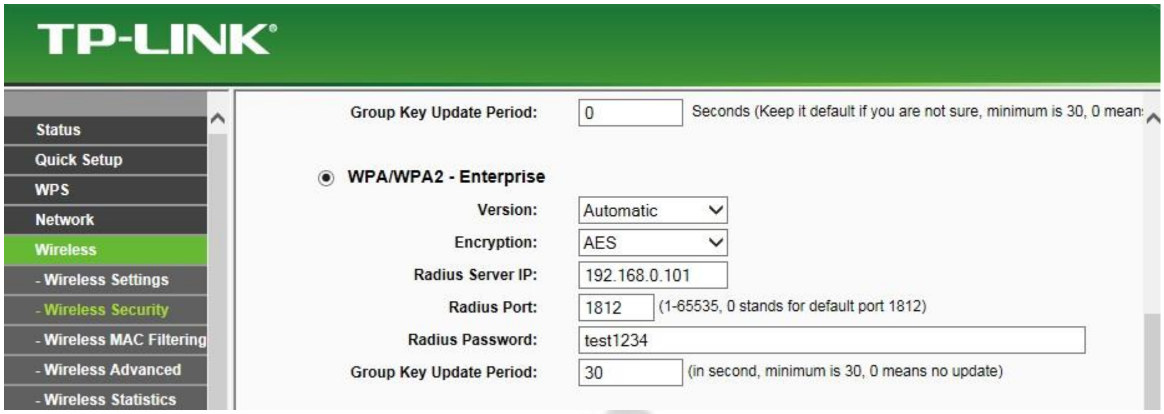


Figure 134: The setting up the AP to support WPA/ WPA2-Enterprise protocol

24. Open the "eap.conf" file from root/usr/loca/etc/raddb and set the default_eap_type to either md5, peap, eap-tls, or eap-ttls. In this example, it is set to "md5" as shown in figure 135. Resave the file.

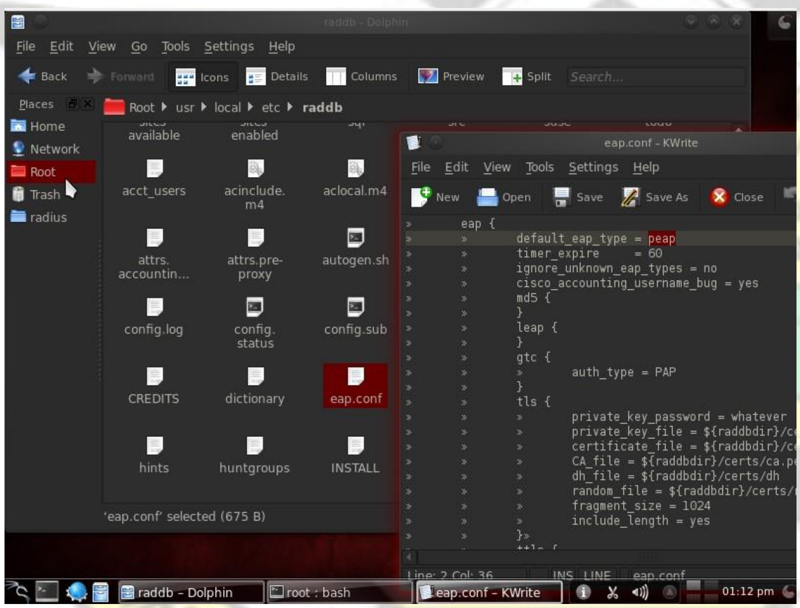


Figure 135: Setting the default_eap_type in the eap.conf file

25. Open the "clients.conf" file from root/usr/loca/etc/raddb and set the shared secret key between the AP and the RADIUS server to the same set in step 21. Our AP is within 192.168.0.0/16 network and its shared secret is "test1234" as shown in figure 136. Re-save the file.

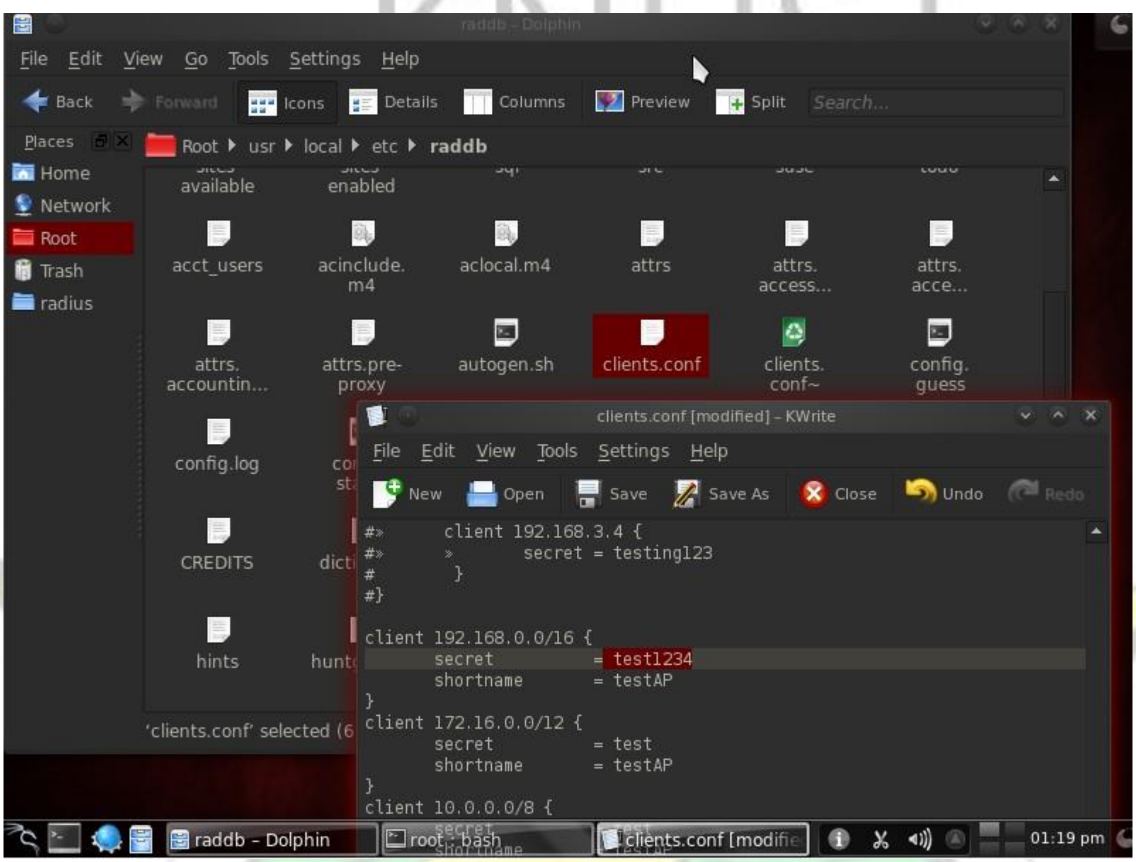


Figure 136: The setting of the secret key between the Radius server and the AP in the client.conf file

26. Select the below and delete from the clients.conf file as shown in figure 137.

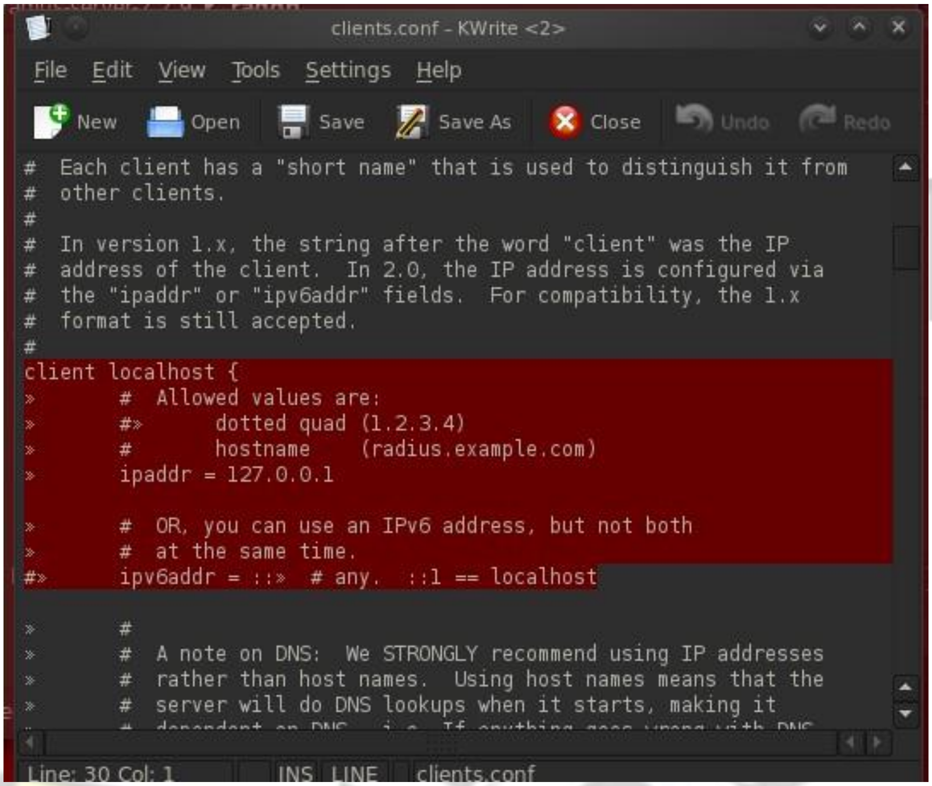


Figure 137: The deletion of the default client localhost portion in the clients.conf

27. Replace it with the below and save. Where the IPaddr is the ip address of the Access Point, the secret is the shared secret key between the AP and the freeradius-wpe server as shown in figure 138. Re-save the file.

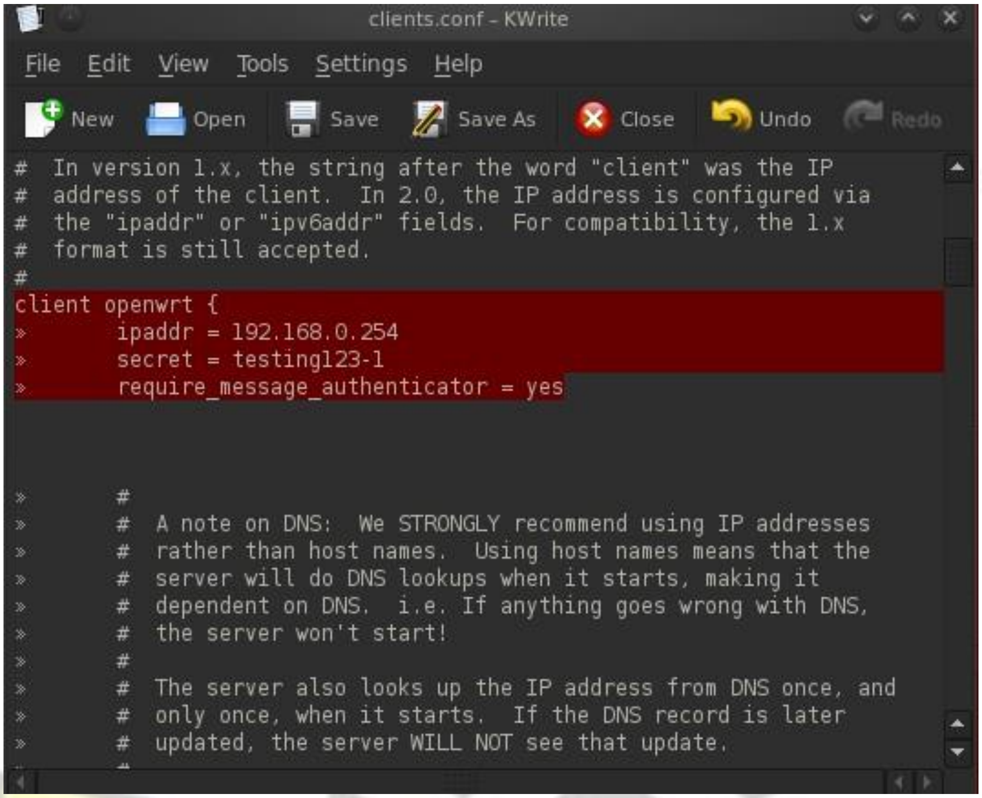


Figure 138: The inclusion of the client openwrt portion in the clients.conf

28. Open the “users” file from root/usr/loca/etc/raddb and set the supplicant username/ password as shown in figure 139. You can set as many users as you want by copying, editing, and repasting “steve>> Cleartext-Password := "testing"” in the user file. Re-save the file.

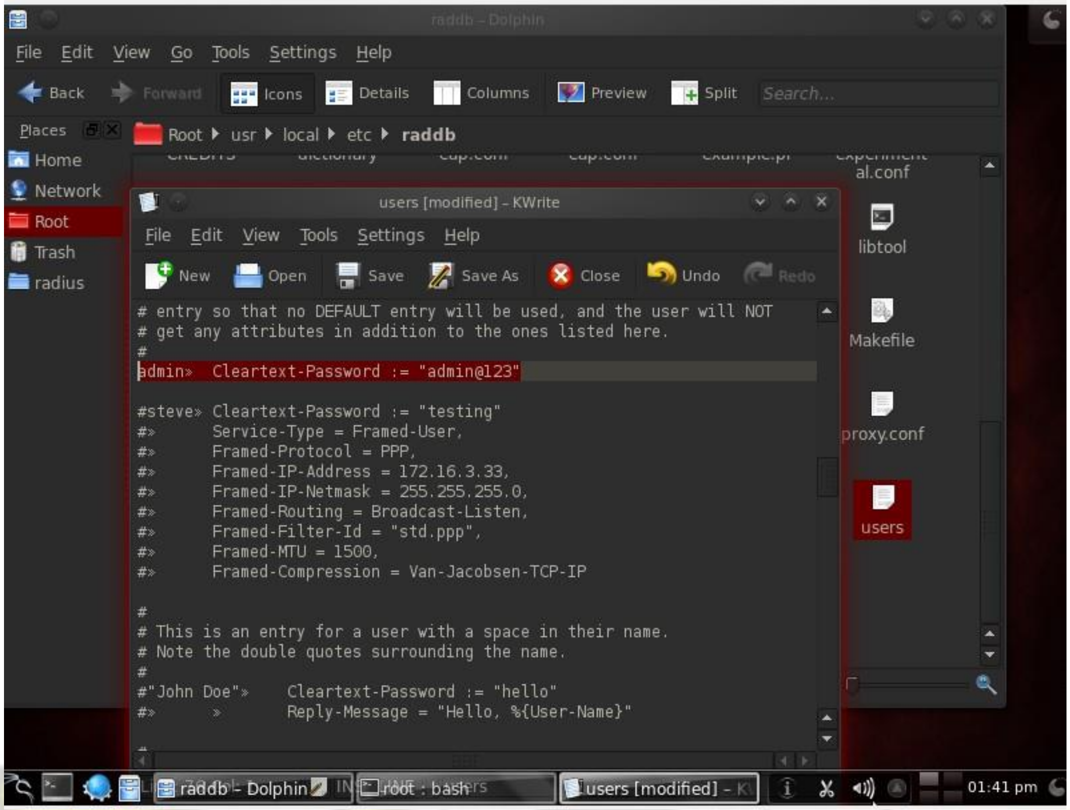


Figure 139: The setting up of the supplicant username/ password in the users file in the freeradius server

29. Start the freeRadius-wpe server with the below command as shown in figure 140.

Figure 140: The results of starting the freeradius-wpe server

30. Configure the supplicant to connect to the Access Point as shown in figure 141.

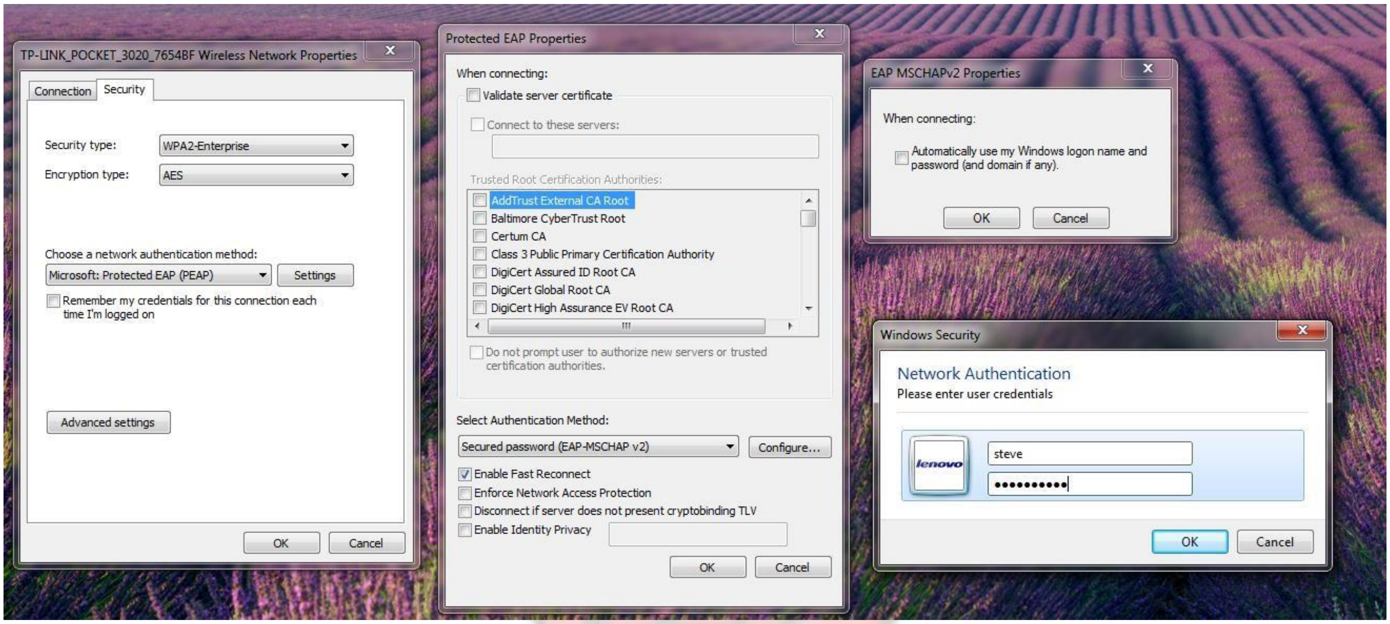


Figure 141: The supplicant configured to support WPA/ WPA-2 Enterprise with same username/password as the authentication server

31. Figure 142 shows a successful authentication of the supplicant to the freeradius-wpe server.

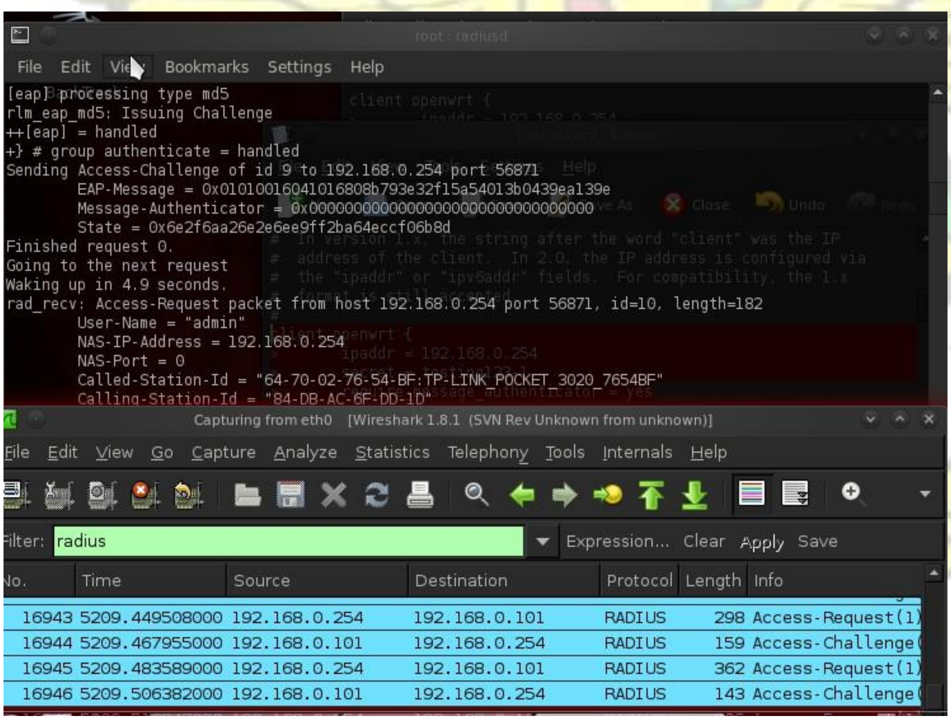


Figure 142: The successful connection between the supplicant and the freeRadius server

Appendix C

Table 4: The below shows the list of the 1,271 Access Points surveyed during the war-driving as discussed in sections 3.11 and 4.10.

Microsoft Excel - WigleWifi_201504051258000

	A	B	C	D	E	F	G	H	I	J	K
1	WigleWifi-1.4	appRelease=2.7	model=HUAWEI MT7-L09	release=4.4.2	device=hwmt7	display=MT7-L09V100R001C00B126	board=MT7-L09	brand=Huawei			
2	MAC	SSID	AuthMode	FirstSeen	Channel	RSSI	CurrentLatitude	CurrentLongitude	AltitudeMeters	AccuracyMeters	Type
3	00:00:00:00:00:00	HoneySuckle_FreeWifi	[ESS]	2/10/2015 15:10	6	-94	5.5687149	-0.18415	49.38645	12	WIFI
4	00:0b:6b:37:af:38	ESSNET_PHQ	[ESS]	2/10/2015 15:09	56	-91	5.56911228	-0.18485	52.62659	16	WIFI
5	00:0c:42:6a:56:c4	Pro2Kaneshie	[ESS]	2/10/2015 14:33	100	-83	5.56648432	-0.22415	42.25751	12	WIFI
6	00:0c:42:6a:94:99	Acc014	[ESS]	2/10/2015 14:35	104	-91	5.56821742	-0.22059	29.15388	12	WIFI
7	00:0c:42:6a:94:a0	AWECG	[ESS]	2/10/2015 14:57	100	-87	5.56856474	-0.2193	39.2039	24	WIFI
8	00:0c:42:a4:c2:8a	iamhe	[ESS]	2/10/2015 14:30	56	-85	5.55434865	-0.23152	40.58132	12	WIFI
9	00:0c:42:f8:f1:51	EDDYS PIZZA	[ESS]	2/10/2015 15:05	1	-67	5.57181212	-0.20634	44.51825	16	WIFI
10	00:10:e7:44:b5:d1	starnetjet	[ESS]	2/10/2015 15:01	5580	-91	5.56922418	-0.21688	39.49832	16	WIFI
11	00:14:f2:7d:ac:70	lutheran_school_Wifi	[ESS]	2/10/2015 15:07	7	-92	5.57436665	-0.19506	32.75837	16	WIFI
12	00:15:62:da:57:00	LaPalm_WiFi	[ESS]	2/10/2015 15:24	4	-96	5.55899653	-0.15087	40.65757	12	WIFI
13	00:15:63:d3:9b:f0	LaPalm_WiFi	[ESS]	2/10/2015 15:24	5	-95	5.55853179	-0.15201	39.53676	12	WIFI
14	00:15:6d:1a:23:15	UGAHSA LINK 1	[ESS]	2/10/2015 14:25	1	-97	5.53651661	-0.22935	52.92135	12	WIFI
15	00:15:6d:64:01:da	ProElectroV	[ESS]	2/10/2015 14:57	48	-91	5.56856474	-0.2193	39.2039	24	WIFI
16	00:15:6d:72:43:db	The-INN-Apartments1	[ESS]	2/10/2015 15:08	1	-87	5.57061568	-0.18755	53.11222	12	WIFI
17	00:15:6d:f6:06:06	PALOMA_WiFi-Zone	[ESS]	2/10/2015 15:04	6	-85	5.57128194	-0.20839	45.4755	16	WIFI
18	00:15:c6:06:fb:40	LaPalm_WiFi	[ESS]	2/10/2015 15:25	8	-90	5.56249437	-0.14473	38.98756	16	WIFI
19	00:1f:27:75:f3:00	BLUECLOUD NETWORKS	[ESS]	2/10/2015 15:08	11	-90	5.57094273	-0.18815	47.50243	16	WIFI
20	00:1f:27:75:f3:02		[ESS]	2/10/2015 15:08	11	-86	5.57079608	-0.1879	51.44039	16	WIFI
21	00:1f:27:75:f3:04		[ESS]	2/10/2015 15:08	11	-90	5.57094273	-0.18815	47.50243	16	WIFI
22	00:21:29:ad:a5:aa	OBSTETRICS WIFI	[ESS]	2/10/2015 14:23	6	-90	5.53733345	-0.22905	40.70858	16	WIFI
23	00:22:0d:70:d0:02		[ESS]	2/10/2015 15:11	11	-93	5.56705922	-0.18123	48.49135	16	WIFI
24	00:25:86:d9:76:a2	GNFSIT2	[ESS]	2/10/2015 15:09	6	-79	5.56915919	-0.18493	51.65226	16	WIFI
25	00:26:cb:6a:ef:40	CIRCLE_LINK	[ESS]	2/10/2015 15:04	1	-91	5.57141072	-0.20787	45.91128	16	WIFI
26	00:26:cb:6a:f1:10	TESHIE_LINK	[ESS]	2/10/2015 15:38	8	-93	5.58783387	-0.09893	36.15541	16	WIFI
27	00:27:22:8a:fe:e6	La Villa Wifi_ G.O	[ESS]	3/18/2015 18:19	1	-75	5.56727196	-0.18348	15.14481	96	WIFI
28	00:27:22:8c:4c:04	BLUESPOT	[ESS]	2/10/2015 15:11	3	-92	5.56628824	-0.17968	49.15025	12	WIFI
29	00:27:22:90:b8:29	CT-SYMPH-AP	[ESS]	2/10/2015 14:30	1	-96	5.55185802	-0.23044	37.91077	12	WIFI
30	00:27:22:c2:10:8f	www.purenetsgh.com	[ESS]	2/10/2015 14:30	8	-92	5.55238904	-0.23067	39.87637	12	WIFI
31	00:27:22:c4:06:3e	PALOMA_WiFi-Zone	[ESS]	2/10/2015 15:05	6	-92	5.57153488	-0.20735	49.33937	16	WIFI
32	00:27:22:c6:9f:ae	YZONE	[ESS]	2/10/2015 15:05	1	-92	5.57201071	-0.20549	50.39645	12	WIFI
33	00:27:22:e2:f7:2b	Pro-DnC	[ESS]	2/10/2015 14:34	1	-96	5.56813697	-0.22083	29.85865	12	WIFI
34	00:27:22:e8:66:92	InertnetGhana{1}	[ESS]	2/10/2015 15:05	1	-92	5.572027	-0.20559	48.92451	12	WIFI
35	00:27:22:e8:66:cd	La Villa Block C	[ESS]	2/10/2015 15:10	1	-80	5.56819563	-0.18325	41.84125	12	WIFI
36	00:30:4f:68:d6:7b		[ESS]	2/10/2015 15:29	2	-93	5.57278399	-0.11414	36.69764	24	WIFI
37	00:50:e8:08:06:84	1	[ESS]	2/10/2015 14:22	6	-89	5.53508611	-0.23152	49.06529	16	WIFI
38	00:60:b3:5d:71:fa	J-prompt-wrenco-AP	[ESS]	2/10/2015 15:01	9	-94	5.56922618	-0.21674	37.55817	8	WIFI
39	00:80:48:5f:0e:93	SIC	[ESS]	2/10/2015 15:10	36	-89	5.56819563	-0.18325	41.84125	12	WIFI
40	02:08:22:f8:ac:4c	AndroidAP	[ESS]	2/10/2015 15:57	1	-82	5.59419628	-0.08879	39.77083	8	WIFI
41	08:3e:8e:02:2e:73	HP-Print-73-LaserJet 1102	[ESS]	2/10/2015 15:16	6	-91	5.55674949	-0.16576	43.34174	16	WIFI
42	1c:3e:84:96:03:dc	HP-Print-DC-LaserJet M1217	[ESS]	2/10/2015 15:06	6	-91	5.5727456	-0.20269	46.37859	16	WIFI
43	20:aa:4b:22:14:db		[ESS]	2/10/2015 15:10	6	-93	5.5687149	-0.18415	49.38645	12	WIFI
44	24:a4:3c:72:bc:c5	NESTOD_0242182573	[ESS]	2/10/2015 16:31	3	-98	5.59469382	-0.0865	59.07143	16	WIFI
45	28:cc:01:99:c5:12	AndroidAP	[ESS]	2/10/2015 14:43	6	-92	5.56841473	-0.21972	37.26252	12	WIFI
46	4c:5e:0c:73:35:31	afol-Dizengoff-ap	[ESS]	2/10/2015 14:56	40	-89	5.56836793	-0.21984	39.98044	16	WIFI
47	4c:5e:0c:c8:7d:e5	afol-nationwide	[ESS]	2/10/2015 14:59	40	-92	5.56874289	-0.21835	33.84633	12	WIFI
48	58:6d:8f:83:a2:a3	OBSTETRICS WIFI	[ESS]	2/10/2015 14:23	6	-96	5.5372416	-0.22913	41.90622	16	WIFI
49	58:6d:8f:9f:78:0b	ROCKPHARM-guest	[ESS]	2/10/2015 14:26	6	-93	5.5360753	-0.22683	50.68963	16	WIFI
50	58:8d:09:e2:d2:ac	BusyInternet HOTSPOT	[ESS]	2/10/2015 15:04	7	-91	5.57113325	-0.20891	47.08502	12	WIFI
51	68:72:51:00:7b:7f	La Villa Block E	[ESS]	3/18/2015 18:19	6	-92	5.56714778	-0.18368	45.47007	64	WIFI
52	80:3f:5d:86:ae:dd	kodako_zip	[ESS]	2/10/2015 15:13	1	-86	5.56251209	-0.17313	43.37319	16	WIFI
53	b8:a3:86:ff:ee:10	dlink1	[ESS]	2/10/2015 14:27	9	-94	5.53610064	-0.2267	50.46413	16	WIFI
54	bc:85:56:4f:a0:01	HP-Print-01-LaserJet 200	[ESS]	2/10/2015 15:06	7	-97	5.5731412	-0.20088	33.02934	16	WIFI

	A	B	C	D	E	F	G	H	I	J	K
1	WigleWifi-1.4	appRelease=2.7	model=HUAWEI MT7-L09	release=4.4.2	device=hwmt7	display=MT7-L09V100R001C00B126	board=MT7-L09	brand=Huawei			
2	MAC	SSID	AuthMode	FirstSeen	Channel	RSSI	CurrentLatitude	CurrentLongitude	AltitudeMeters	AccuracyMeters	Type
3	bc:85:56:4f:aa:a2	HP-Print-a2-AFOL WIRELESS P	[ESS]	2/10/2015 15:02	6	-94	5.56927753	-0.21637	34.86557	12	WIFI
4	c8:d3:a3:2c:8b:4a	Labadi Beach Hotel - Wifi	[ESS]	2/10/2015 15:26	6	-92	5.56468914	-0.13747	37.89295	12	WIFI
5	d2:b3:3f:e5:05:a7	ADYYnVzb2xh	[ESS]	2/10/2015 15:06	1	-93	5.57294911	-0.20178	41.67327	24	WIFI
6	d4:ca:6d:50:a9:45	afol-emiratesHQ	[ESS]	2/10/2015 15:02	40	-91	5.56931136	-0.21514	30.83431	16	WIFI
7	d4:ca:6d:50:a9:8f	afol-mecoy-ap	[ESS]	2/10/2015 14:35	56	-93	5.56830894	-0.22042	31.8088	12	WIFI
8	d4:ca:6d:ce:ab:85	afol-mantrac-vl	[ESS]	2/10/2015 14:35	36	-89	5.56833003	-0.22038	32.23633	12	WIFI
9	dc:9f:db:0a:4f:2b	PALOMA_WiFi-Zone	[ESS]	2/10/2015 15:05	6	-90	5.57207802	-0.20523	50.14638	16	WIFI
10	dc:9f:db:54:da:43	S02_2G_BTS	[ESS]	2/10/2015 15:09	5	-90	5.57008869	-0.18658	47.33652	16	WIFI
11	dc:9f:db:5a:c9:0b	OPS-Wifi	[ESS]	2/10/2015 14:35	6	-90	5.56834689	-0.22031	33.22935	8	WIFI
12	dc:9f:db:63:28:a4		[ESS]	2/10/2015 15:05	1	-94	5.57200929	-0.20566	48.88391	16	WIFI
13	dc:9f:db:6c:02:10	LOVCATHEDRAL	[ESS]	2/10/2015 15:30	1	-93	5.57880678	-0.10937	42.90189	12	WIFI
14	dc:9f:db:6c:04:1b	LOVCATHEDRAL	[ESS]	2/10/2015 15:30	1	-97	5.57880678	-0.10937	42.90189	12	WIFI
15	dc:9f:db:76:7c:09	ALK_BASE4	[ESS]	2/10/2015 15:05	40	-90	5.572027	-0.20559	48.88148	24	WIFI
16	dc:9f:db:9c:0e:65	LOVCATHEDRAL	[ESS]	2/10/2015 15:30	1	-96	5.57847276	-0.10964	40.23556	12	WIFI
17	f8:d1:11:81:c9:32	UGAHSA HTSPT B WING	[ESS]	2/10/2015 14:22	11	-96	5.53555969	-0.22925	40.38616	12	WIFI
18	fa:8f:ca:6c:b4:ef	venice_FAFI	[ESS]	2/10/2015 15:05	1	-95	5.57212765	-0.20492	45.8138	16	WIFI
19	00:0c:42:39:11:85	Arrow-WWW	[WEP][ESS]	2/10/2015 15:01	120	-89	5.56922418	-0.21688	39.49832	16	WIFI
20	00:0c:42:66:26:b8	ArrowRR	[WEP][ESS]	2/10/2015 15:02	140	-91	5.5693892	-0.21507	32.25602	12	WIFI
21	00:27:19:cb:20:da	Vizico	[WEP][ESS]	2/10/2015 15:39	5	-95	5.58859937	-0.09765	42.49541	12	WIFI
22	00:e0:4d:cf:d0:8a	HG520c	[WEP][ESS]	2/10/2015 14:22	1	-94	5.53508611	-0.23152	49.06529	16	WIFI
23	24:a4:3c:a6:66:af	Angutech Wireless	[WEP][ESS]	2/10/2015 15:40	10	-93	5.59060154	-0.09429	40.63089	24	WIFI
24	48:f8:b3:c3:c5:cf	Linksys01313	[WEP][ESS]	2/10/2015 15:04	4	-88	5.5703827	-0.21118	30.84723	16	WIFI
25	bc:76:70:66:d0:14	Vodafone HG530 Home Gatew	[WEP][ESS]	2/10/2015 15:35	1	-94	5.58468923	-0.10291	38.13597	16	WIFI
26	c8:3a:35:3e:ac:f0	Tenda	[WEP][ESS]	2/10/2015 14:33	1	-94	5.56505676	-0.22568	42.63473	16	WIFI
27	d4:ca:6d:8c:1d:1f	BivacGCB	[WEP][ESS]	2/10/2015 14:56	100	-92	5.56849229	-0.21961	39.07004	24	WIFI
28	f4:ec:38:fa:27:a3	lebanonemb	[WEP][ESS]	2/10/2015 15:10	6	-94	5.56770063	-0.18235	45.12143	16	WIFI
29	c4:14:3c:38:69:70	CorpNet	[WPA2-EAP-CCMP][ESS]	2/10/2015 14:34	1	-95	5.56788254	-0.22224	38.71305	12	WIFI
30	00:01:23:45:67:89	MMKLTD	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:06	6	-93	5.57294911	-0.20178	41.67327	24	WIFI
31	00:18:25:03:3a:b0	pub_modevc_S1	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:07	7	-92	5.57424726	-0.19432	36.54047	24	WIFI
32	00:18:25:10:b3:20	WAVION_BS	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:10	6	-85	5.56773974	-0.18242	45.22401	16	WIFI
33	00:23:89:d2:ad:70	COMMUNICATION	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:08	1	-94	5.57097499	-0.18822	47.24717	12	WIFI
34	00:27:22:c8:72:f4	sronko1	[WPA2-PSK-CCMP][ESS]	2/10/2015 14:31	4	-93	5.55970113	-0.23023	38.84518	12	WIFI
35	24:a4:3c:72:be:89	Ecoband_Osu	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:11	7	-93	5.56669925	-0.18044	50.17347	16	WIFI
36	24:a4:3c:ac:f4:72	ISIS 2	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:54	6	-95	5.60125556	-0.09277	44.57008	8	WIFI
37	24:db:ac:47:f1:14	VodafoneMobileWiFi-F11428	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:13	1	-93	5.56238545	-0.17288	45.22215	16	WIFI
38	24:db:ac:8c:1d:7e	VodafoneMobileWiFi-1D7E96	[WPA2-PSK-CCMP][ESS]	2/10/2015 14:27	1	-95	5.53644067	-0.22517	50.28102	24	WIFI
39	3c:e5:a6:12:74:40	POLICE_CID	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:08	1	-91	5.57069381	-0.18772	52.68102	16	WIFI
40	3c:e5:a6:12:74:41	POLICE_CID_1	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:08	1	-94	5.57043374	-0.18726	50.87045	12	WIFI
41	40:0e:85:f3:29:93	AndroidHotspot3314	[WPA2-PSK-CCMP][ESS]	2/10/2015 14:32	6	-91	5.56017093	-0.22996	43.84572	8	WIFI
42	50:9f:27:6f:6d:18	INNOCENT	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:23	6	-92	5.55802005	-0.15875	44.65531	12	WIFI
43	50:9f:27:6f:74:83	VodafoneMobileWiFi-748357	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:04	1	-91	5.57052918	-0.21065	37.56858	16	WIFI
44	50:9f:27:6f:78:bf	VodafoneMobileWiFi-78BF74	[WPA2-PSK-CCMP][ESS]	2/10/2015 14:22	6	-92	5.53525072	-0.23077	38.48704	12	WIFI
45	50:9f:27:6f:7f:d3	VodafoneMobileWiFi-7FD332	[WPA2-PSK-CCMP][ESS]	2/10/2015 14:22	6	-93	5.53564482	-0.2292	40.9033	8	WIFI
46	50:9f:27:6f:87:28	VodafoneMobileWiFi-872867	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:57	6	-93	5.59490226	-0.08908	47.82179	12	WIFI
47	50:9f:27:6f:8b:14	Izzy Taylor-Link	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:03	11	-86	5.5694794	-0.21497	34.17788	12	WIFI
48	58:1f:aa:e6:ea:2b	Kofi B's Wi-Fi Network	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:04	6	-89	5.57052918	-0.21065	37.56858	16	WIFI
49	76:e4:3a:8b:5d:7d	MOSES	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:38	1	-97	5.58791675	-0.09876	36.76103	12	WIFI
50	78:54:2e:56:76:1a	FGBMFI-AccessPOINT	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:12	1	-90	5.56273487	-0.17351	38.81092	16	WIFI
51	7c:d1:c3:cd:2a:d0	LPS-Music	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:11	13	-82	5.56727154	-0.18152	50.07831	16	WIFI
52	80:57:19:48:bd:4d	MTA	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:13	1	-93	5.56251209	-0.17313	43.37319	16	WIFI
53	84:51:81:83:2b:e8	AndroidAP	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:37	6	-90	5.58677781	-0.10048	39.67097	12	WIFI
54	84:7a:88:17:12:eb	AHTC Portable Hotspot 896D	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:01	6	-94	5.5691664	-0.21707	39.619	16	WIFI
55	84:a8:e4:1b:15:90	Matrix	[WPA2-PSK-CCMP][ESS]	2/10/2015 14:21	1	-92	5.53456663	-0.23401	44.60291	12	WIFI
56	90:72:40:0e:f0:2a	FinGAP's Wi-Fi	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:08	6	-88	5.57061568	-0.18755	53.11222	12	WIFI
57	9c:3a:af:d9:b6:1e	phil wifi	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:27	6	-92	5.56519725	-0.13616	41.35274	12	WIFI

	A	B	C	D	E	F	G	H	I	J	K
1	WigleWifi-1.4	appRelease=2.7	model=HUAWEI MT7-L09	release=4.4.2	device=hwmt7	display=MT7-L09V100R001C00B126	board=MT7-L09	brand=Huawei			
2	MAC	SSID	AuthMode	FirstSeen	Channel	RSSI	CurrentLatitude	CurrentLongitude	AltitudeMeters	AccuracyMeters	Type
3	b0:df:3a:fd:19:c1	AndroidAP	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:09	6	-81	5.5695797	-0.18566	44.94277	16	WIFI
4	bc:79:ad:a0:ca:7b	rukia labouja	[WPA2-PSK-CCMP][ESS]	2/10/2015 14:33	6	-92	5.56706828	-0.2235	39.71772	12	WIFI
5	c0:4a:00:60:52:4a	TRV-COMSYS	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:09	8	-91	5.56935671	-0.18525	48.4254	24	WIFI
6	c4:62:ea:74:86:5b	AndroidAP	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:08	6	-85	5.57069381	-0.18772	52.68102	16	WIFI
7	cc:a2:23:88:6e:96	VodafoneMobileWiFi-6E9621	[WPA2-PSK-CCMP][ESS]	2/10/2015 14:58	6	-90	5.5686312	-0.21897	33.67424	12	WIFI
8	cc:a2:23:88:75:10	VodafoneMobileWiFi-751079	[WPA2-PSK-CCMP][ESS]	2/10/2015 14:38	6	-96	5.56837832	-0.21984	36.36511	16	WIFI
9	cc:f3:a5:46:be:90	zikay.net	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:37	6	-92	5.58773452	-0.09912	36.0551	12	WIFI
10	cc:f3:a5:46:be:90	zikay.net	[WPA2-PSK-CCMP][ESS]	2/10/2015 16:21	6	-94	5.59477557	-0.08643	51.62112	12	WIFI
11	cc:f9:e8:a1:f8:7f	AndroidAP7588	[WPA2-PSK-CCMP][ESS]	2/10/2015 14:28	1	-86	5.53768872	-0.22176	49.93014	12	WIFI
12	e6:f4:c6:02:d3:98	Esoko-Guest	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:04	1	-93	5.57033497	-0.21167	25.9256	24	WIFI
13	f0:25:b7:66:a0:fe	ihab	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:07	1	-95	5.57413391	-0.19675	38.20125	16	WIFI
14	f0:5a:09:29:e3:95	AndroidAP7429	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:15	6	-92	5.55605057	-0.16761	42.57668	12	WIFI
15	f0:79:59:27:e3:94	My ASUS	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:04	6	-93	5.57052918	-0.21065	37.56858	16	WIFI
16	f4:1f:c2:d0:6f:90	Mantrac-Accra	[WPA2-PSK-CCMP][ESS]	2/10/2015 14:35	6	-97	5.56830894	-0.22042	31.8088	12	WIFI
17	f4:9f:f3:8d:d3:2e	VodafoneMobileWiFi-D32E54	[WPA2-PSK-CCMP][ESS]	2/10/2015 14:27	1	-87	5.53669224	-0.22394	52.72159	12	WIFI
18	f4:9f:f3:95:4e:72	VodafoneMobileWiFi-4E7282	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:09	1	-89	5.56911243	-0.18485	52.67367	16	WIFI
19	f4:9f:f3:95:52:b6	VodafoneMobileWiFi-52B674	[WPA2-PSK-CCMP][ESS]	2/10/2015 15:24	4	-89	5.55890824	-0.15109	39.58044	16	WIFI
20	00:14:d1:b1:36:6a	Africa Online	[WPA2-PSK-CCMP][WPS][ESS]	2/10/2015 15:02	9	-92	5.56930556	-0.21589	38.24923	16	WIFI
21	00:73:05:04:0e:cc	iPush-LN-040ECC	[WPA2-PSK-CCMP][WPS][ESS]	2/10/2015 14:22	6	-94	5.53525072	-0.23077	38.48704	12	WIFI
22	02:08:22:04:0d:01	nii kotei	[WPA2-PSK-CCMP][WPS][ESS]	2/10/2015 15:10	1	-91	5.5686276	-0.184	46.94587	16	WIFI
23	02:0c:e7:43:83:87	Infinix X403	[WPA2-PSK-CCMP][WPS][ESS]	2/10/2015 15:06	1	-91	5.57294911	-0.20178	41.67327	24	WIFI
24	48:f8:b3:21:4d:01	M-Solutions	[WPA2-PSK-CCMP][WPS][ESS]	2/10/2015 15:07	1	-90	5.57413391	-0.19675	38.20125	16	WIFI
25	58:6d:8f:19:bf:06	Provident-Life	[WPA2-PSK-CCMP][WPS][ESS]	2/10/2015 15:04	1	-89	5.57033497	-0.21167	25.9256	24	WIFI
26	58:6d:8f:91:d3:8c	Millitech Limited	[WPA2-PSK-CCMP][WPS][ESS]	2/10/2015 14:30	6	-91	5.55528006	-0.23175	29.33292	12	WIFI
27	60:51:2c:5d:66:8e	Tigo Internet_668E	[WPA2-PSK-CCMP][WPS][ESS]	2/10/2015 14:22	4	-96	5.53508611	-0.23152	49.06529	16	WIFI
28	64:70:02:f1:8f:e8	GRAOSUSTO	[WPA2-PSK-CCMP][WPS][ESS]	2/10/2015 15:08	7	-93	5.57207485	-0.19034	48.55961	32	WIFI
29	bc:77:37:89:8a:a1	FREDERICKA-PC-05630	[WPA2-PSK-CCMP][WPS][ESS]	2/10/2015 15:06	1	-98	5.57284662	-0.20223	45.28488	16	WIFI
30	c8:3a:35:20:d7:b0	GOLDEN CEDI	[WPA2-PSK-CCMP][WPS][ESS]	2/10/2015 14:32	6	-94	5.55997252	-0.23008	41.72316	12	WIFI
31	c8:d7:19:e4:e9:b1	DOXACAPITAL	[WPA2-PSK-CCMP][WPS][ESS]	2/10/2015 15:04	1	-94	5.57098183	-0.20945	45.15221	16	WIFI
32	e4:f4:c6:02:d3:97	Esoko	[WPA2-PSK-CCMP][WPS][ESS]	2/10/2015 15:04	1	-93	5.5703827	-0.21118	30.84723	16	WIFI
33	00:18:e7:8d:32:3a	sic_wireless	[WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:10	1	-97	5.56852675	-0.18383	45.0141	16	WIFI
34	00:1f:27:75:f3:01		[WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:08	11	-90	5.57097499	-0.18822	47.24717	12	WIFI
35	00:27:22:12:f7:5c	MMTLink	[WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 14:32	1	-94	5.56063316	-0.22967	43.72269	12	WIFI
36	00:e0:4c:f2:0f:f0	A-Plus WIFI	[WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:12	6	-82	5.56348698	-0.17476	47.80522	24	WIFI
37	20:aa:4b:99:7b:0a	CHILDWIRE	[WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 14:26	6	-89	5.53581429	-0.22804	55.22647	16	WIFI
38	48:f8:b3:3b:a2:83	PHARMACY DEPT	[WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 14:27	6	-90	5.53631958	-0.22581	49.4306	16	WIFI
39	50:67:ae:c3:22:b0	Mantrac-WH	[WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 14:34	4	-93	5.56800169	-0.22131	36.59896	12	WIFI
40	78:6a:89:43:5a:37	WLAN1-460060507000238	[WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 14:32	6	-95	5.56017093	-0.22996	43.84572	8	WIFI
41	88:ce:fa:6c:6f:95	Surfline4GLTE6F95	[WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:42	1	-91	5.5970813	-0.09585	41.09201	12	WIFI
42	88:ce:fa:6c:76:70	Surfline4GLTE7670	[WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:39	1	-95	5.589191	-0.09666	49.3863	24	WIFI
43	88:ce:fa:6c:99:e8	Surfline4GLTE99E8	[WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:11	8	-86	5.56727154	-0.18152	50.07831	16	WIFI
44	88:ce:fa:6c:9a:60	MokatSurfline	[WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:16	1	-95	5.55671131	-0.16571	43.00164	12	WIFI
45	dc:9f:db:02:97:a0	WestOne1F	[WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:11	11	-87	5.56618599	-0.1795	46.6305	16	WIFI
46	dc:9f:db:3c:a2:c7	KAMA	[WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:13	11	-85	5.561422	-0.17092	53.32956	16	WIFI
47	e8:08:8b:8b:ea:54	SeniorWifi	[WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:11	9	-97	5.56602382	-0.17921	42.76491	16	WIFI
48	f4:dc:f9:30:ae:ea	Surfline4GLTEAEEA	[WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:11	5	-93	5.56648676	-0.18001	51.27197	24	WIFI
49	f4:dc:f9:30:b5:57	Surfline4GLTEB557	[WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:04	1	-94	5.57113325	-0.20891	47.08502	12	WIFI
50	f4:dc:f9:30:c0:92	Surfline4GLTEC092	[WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:05	1	-92	5.57221773	-0.20455	43.01228	12	WIFI
51	f4:dc:f9:30:cf:ba	Surfline4GLTECFBA	[WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:57	1	-85	5.59490226	-0.08908	47.82179	12	WIFI
52	f4:dc:f9:30:d2:76	Surfline4GLTED276	[WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 14:24	1	-96	5.53734532	-0.22886	42.58552	12	WIFI
53	f4:dc:f9:30:d2:85	Surfline4GLTED285	[WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:41	4	-91	5.59563547	-0.09526	40.59658	12	WIFI
54	f4:dc:f9:30:d9:10	Surfline4G	[WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:13	1	-86	5.56119482	-0.17043	58.13226	24	WIFI
55	00:22:6b:f5:c8:69	FGBMFI REPEATER	[WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:13	11	-81	5.56206997	-0.17237	47.38768	16	WIFI
56	74:5c:9f:4c:0a:0c	Surfline4GLTE_0A0C	[WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:54	11	-92	5.60108803	-0.09319	43.06847	12	WIFI
57	74:5c:9f:4c:0a:3a	Surfline4GLTE_0A3A	[WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:06	6	-96	5.57260096	-0.20318	45.603	16	WIFI

	A	B	C	D	E	F	G	H	I	J	K
1	WigleWifi-1.4	appRelease=2.7	model=HUAWEI MT7-L09	release=4.4.2	device=hwmt7	display=MT7-L09V100R001C00B126	board=MT7-L09	brand=Huawei			
2	MAC	SSID	AuthMode	FirstSeen	Channel	RSSI	CurrentLatitude	CurrentLongitude	AltitudeMeters	AccuracyMeters	Type
3	74:5c:9f:4c:0a:71	Elvisizzles	[WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:04	1	-89	5.5703232	-0.21211	26.64215	16	WIFI
4	74:5c:9f:4c:0b:77	Fifth Side	[WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 14:30	6	-90	5.54943618	-0.22938	34.69377	16	WIFI
5	74:5c:9f:4c:0d:17	Surfline4GLTE_0D17	[WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:22	6	-94	5.55712702	-0.16474	48.25722	8	WIFI
6	a0:f3:c1:48:54:f2	WBE-MANAGER	[WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:39	1	-94	5.58838095	-0.098	40.155	12	WIFI
7	e8:de:27:2f:d9:f4	ITTAS_WIFI_04	[WPA2-PSK-CCMP+TKIP][WPS][ESS]	3/18/2015 18:19	9	-89	5.56727196	-0.18348	15.14481	96	WIFI
8	c8:3a:35:02:0d:e8	ASIP_MAIN_OFFICE_1	[WPA2-PSK-CCMP+TKIP-preauth][WPS][ESS]	2/10/2015 14:34	5	-92	5.56784114	-0.22239	38.86067	16	WIFI
9	00:50:e8:08:06:85	2	[WPA2-PSK-CCMP-preauth][ESS]	2/10/2015 14:22	6	-89	5.53508611	-0.23152	49.06529	16	WIFI
10	00:50:e8:08:06:86	3	[WPA2-PSK-CCMP-preauth][ESS]	2/10/2015 14:22	6	-89	5.53508611	-0.23152	49.06529	16	WIFI
11	f8:1a:67:bb:8d:fc	WESTEC-WIFI	[WPA2-PSK-CCMP-preauth][ESS]	2/10/2015 15:04	6	-94	5.57128194	-0.20839	45.4755	16	WIFI
12	68:7f:74:f1:87:13	TOPTECH-W	[WPA2-PSK-CCMP-preauth][WPS][ESS]	2/10/2015 15:15	6	-95	5.55674639	-0.16583	43.6347	16	WIFI
13	00:27:22:fa:17:11	GAFCSC-HQ	[WPA2-PSK-TKIP][ESS]	2/10/2015 15:29	11	-86	5.57463425	-0.11202	47.28997	12	WIFI
14	00:27:22:fa:18:0f	JUNIOR DIVISION	[WPA2-PSK-TKIP][ESS]	2/10/2015 15:29	5	-86	5.57149119	-0.11628	40.57627	24	WIFI
15	24:a4:3c:44:ed:bb	PCG EBENEZER COMPOUND	[WPA2-PSK-TKIP][ESS]	2/10/2015 14:21	6	-95	5.5340839	-0.23607	38.62149	8	WIFI
16	2a:a4:3c:4f:1f:4b	TS-Wifi	[WPA2-PSK-TKIP][ESS]	2/10/2015 15:06	6	-91	5.57260096	-0.20318	45.603	16	WIFI
17	84:a8:e4:1a:f0:20	waxadmin	[WPA2-PSK-TKIP][ESS]	2/10/2015 14:19	1	-94	5.53369479	-0.23768	47.145	12	WIFI
18	84:a8:e4:1b:32:08	UNIROB	[WPA2-PSK-TKIP][ESS]	2/10/2015 14:27	6	-90	5.5365243	-0.22475	52.37365	16	WIFI
19	84:a8:e4:1b:35:4c	Blue bowl	[WPA2-PSK-TKIP][ESS]	2/10/2015 14:26	1	-95	5.53589023	-0.22766	51.18589	12	WIFI
20	84:a8:e4:1b:38:70	XpressGraphics	[WPA2-PSK-TKIP][ESS]	2/10/2015 14:22	1	-87	5.53513447	-0.23131	48.76327	16	WIFI
21	84:a8:e4:1b:3a:e8	Admin	[WPA2-PSK-TKIP][ESS]	2/10/2015 15:13	6	-87	5.56119482	-0.17043	58.13226	24	WIFI
22	c8:d5:fe:1e:32:c0	win seal	[WPA2-PSK-TKIP][ESS]	2/10/2015 15:31	1	-94	5.57970324	-0.10846	39.30422	12	WIFI
23	cc:96:a0:da:26:60	OwusuNetwork	[WPA2-PSK-TKIP][ESS]	2/10/2015 15:40	1	-93	5.5903391	-0.0947	36.74658	16	WIFI
24	dc:9f:db:08:37:d5	AP_FCN_INT_REL_01	[WPA2-PSK-TKIP][ESS]	2/10/2015 15:08	13	-75	5.57043374	-0.18726	50.87045	12	WIFI
25	f8:d1:11:2a:c7:9a	Alliance_Motors	[WPA2-PSK-TKIP][WPS][ESS]	2/10/2015 14:33	1	-93	5.56306048	-0.22795	40.74657	12	WIFI
26	00:0c:42:26:2e:ba	richfield-0244368642	[WPA-PSK-CCMP][ESS]	2/10/2015 15:08	6	-96	5.57138879	-0.18902	46.21096	16	WIFI
27	00:15:6d:72:38:c0	PUCGS1	[WPA-PSK-CCMP][ESS]	2/10/2015 15:13	1	-89	5.561422	-0.17092	53.32956	16	WIFI
28	00:27:22:fa:17:73	GAFCSC-LIBRARY	[WPA-PSK-CCMP][ESS]	2/10/2015 15:29	10	-92	5.57408126	-0.11244	44.00741	16	WIFI
29	00:27:22:fa:17:bf	GAFCSC-ANKRAH	[WPA-PSK-CCMP][ESS]	2/10/2015 15:29	2	-95	5.57322698	-0.11346	36.79532	16	WIFI
30	64:66:b3:a4:10:ec	KEY AP	[WPA-PSK-CCMP][ESS]	2/10/2015 15:12	5	-90	5.56398033	-0.17554	44.17317	16	WIFI
31	00:15:6d:63:e9:e8	Ssnit-GH	[WPA-PSK-CCMP][WPA2-PSK-CCMP][ESS]	2/10/2015 15:08	5	-91	5.57094273	-0.18815	47.50243	16	WIFI
32	4c:5e:0c:73:c2:ee	PCGH-Alnk-PT	[WPA-PSK-CCMP][WPA2-PSK-CCMP][ESS]	2/10/2015 14:29	3	-95	5.54157485	-0.22551	44.03404	16	WIFI
33	d0:7a:b5:2d:83:8d	AndroidAP	[WPA-PSK-CCMP][WPA2-PSK-CCMP][ESS]	2/10/2015 14:22	1	-89	5.53525072	-0.23077	38.48704	12	WIFI
34	08:63:61:c4:94:1c	Precom	[WPA-PSK-CCMP][WPA2-PSK-CCMP][WPS][ESS]	2/10/2015 15:31	11	-82	5.58115333	-0.10687	40.6228	12	WIFI
35	20:a9:9b:ec:80:5d	Nokia_XL	[WPA-PSK-CCMP][WPA2-PSK-CCMP][WPS][ESS]	2/10/2015 15:57	1	-94	5.59411499	-0.08874	39.35269	12	WIFI
36	64:66:b3:54:8e:24	Hyundai_Conf	[WPA-PSK-CCMP][WPA2-PSK-CCMP][WPS][ESS]	2/10/2015 14:32	10	-85	5.56194109	-0.22888	40.55934	16	WIFI
37	90:f6:52:af:92:6a	WES_SECURITY	[WPA-PSK-CCMP][WPA2-PSK-CCMP][WPS][ESS]	2/10/2015 15:04	36	-91	5.57128194	-0.20839	45.4755	16	WIFI
38	e8:de:27:49:89:4e	NEL SUPPLIES LTD	[WPA-PSK-CCMP][WPA2-PSK-CCMP][WPS][ESS]	2/10/2015 15:05	8	-89	5.57201071	-0.20549	50.39645	12	WIFI
39	e8:de:27:59:3a:26	Mr. President	[WPA-PSK-CCMP][WPA2-PSK-CCMP][WPS][ESS]	2/10/2015 15:12	6	-85	5.56439272	-0.1762	31.2838	16	WIFI
40	ec:17:2f:b5:2f:12	TP-LINK_B52F12	[WPA-PSK-CCMP][WPA2-PSK-CCMP][WPS][ESS]	2/10/2015 14:26	6	-86	5.53581429	-0.22804	55.22647	16	WIFI
41	08:7a:4c:51:a9:9c	VDF-HG532e-CATDQQ	[WPA-PSK-CCMP][WPS][ESS]	2/10/2015 14:33	1	-89	5.56505676	-0.22568	42.63473	16	WIFI
42	90:5f:2e:89:1f:29	Y580E_1F29	[WPA-PSK-CCMP][WPS][ESS]	2/10/2015 15:04	2	-92	5.57033497	-0.21167	25.9256	24	WIFI
43	d4:6e:5c:ec:27:e4	OHWO	[WPA-PSK-CCMP][WPS][ESS]	2/10/2015 15:41	1	-88	5.59425664	-0.09473	41.38582	16	WIFI
44	e8:de:27:5f:55:3c	AAL GHANA	[WPA-PSK-CCMP][WPS][ESS]	2/10/2015 15:04	6	-91	5.57098183	-0.20945	45.15221	16	WIFI
45	00:27:22:4c:cf:3a	RESURGE-AFRICA	[WPA-PSK-CCMP+TKIP][ESS]	2/10/2015 14:24	1	-92	5.53729248	-0.22891	41.27737	12	WIFI
46	00:27:22:94:32:8b	MAGNA GROUP	[WPA-PSK-CCMP+TKIP][ESS]	2/10/2015 15:06	1	-88	5.57260096	-0.20318	45.603	16	WIFI
47	68:72:51:06:07:5e	N.N EST MET. COMP.	[WPA-PSK-CCMP+TKIP][ESS]	2/10/2015 14:33	1	-95	5.56467714	-0.22607	40.872	12	WIFI
48	76:5c:9f:93:66:03	SAPPS_Vodafone Smart Tab 3	[WPA-PSK-CCMP+TKIP][ESS]	2/10/2015 14:35	6	-89	5.56833282	-0.22037	32.36435	16	WIFI
49	94:d7:71:e0:63:d0	Galaxy_Express_5910	[WPA-PSK-CCMP+TKIP][ESS]	2/10/2015 15:25	6	-87	5.56155292	-0.14609	36.26293	16	WIFI
50	00:13:5e:51:c4:25	ZainInternet	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:05	6	-93	5.57153488	-0.20735	49.33937	16	WIFI
51	00:24:01:f2:9f:6a	TAP-WIFI	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:14	2	-94	5.55699738	-0.16744	42.91235	12	WIFI
52	00:26:5a:a2:29:fa	First Floor	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:09	1	-87	5.57008869	-0.18658	47.33652	16	WIFI
53	00:e0:4c:f4:75:54	copyprint	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 14:22	6	-89	5.53508611	-0.23152	49.06529	16	WIFI
54	00:e0:4d:70:41:60	Happytour	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:15	10	-87	5.55661005	-0.16619	43.02736	24	WIFI
55	00:e0:4d:cf:ee:4a	Puh-bear	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:15	1	-95	5.55569488	-0.16892	44.82033	16	WIFI
56	0a:18:d6:5f:af:aa	Starrfm	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:05	1	-98	5.57194746	-0.20593	45.92768	24	WIFI
57	0a:18:d6:5f:b1:d0	Starrfm	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:06	11	-74	5.57294911	-0.20178	41.67327	24	WIFI

	A	B	C	D	E	F	G	H	I	J	K
1	WigleWifi-1.4	appRelease=2.7	model=HUAWEI MT7-L09	release=4.4.2	device=hwmt7	display=MT7-L09V100R001C00B126	board=MT7-L09	brand=Huawei			
2	MAC	SSID	AuthMode	FirstSeen	Channel	RSSI	CurrentLatitude	CurrentLongitude	AltitudeMeters	AccuracyMeters	Type
3	0a:18:d6:5f:b1:d9	Starrfm	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:06	6	-94	5.57284662	-0.20223	45.28488	16	WIFI
4	14:d6:4d:b1:11:04	Elearning-1	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 14:23	9	-90	5.53731171	-0.22896	41.0688	24	WIFI
5	2a:a4:3c:83:5f:23	Accent-down	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:04	6	-84	5.57052918	-0.21065	37.56858	16	WIFI
6	2a:a4:3c:85:e3:7e	Accent-down	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:04	6	-84	5.57052918	-0.21065	37.56858	16	WIFI
7	2e:a4:3c:83:5f:23	Accent-UP	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:04	6	-80	5.57052918	-0.21065	37.56858	16	WIFI
8	2e:a4:3c:85:e3:7e	Accent-UP	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:04	6	-80	5.57052918	-0.21065	37.56858	16	WIFI
9	30:f3:1d:17:b8:9a	AIRTEL-RINGROAD	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:06	1	-91	5.57284662	-0.20223	45.28488	16	WIFI
10	48:f8:b3:48:41:07	AFTPLINK	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:04	6	-88	5.57098183	-0.20945	45.15221	16	WIFI
11	54:e6:fc:cf:8f:80	CometGH	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:06	9	-95	5.5727456	-0.20269	46.37859	16	WIFI
12	64:70:02:7d:f5:9c	ALBRIM	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:15	1	-89	5.55661005	-0.16619	43.02736	24	WIFI
13	68:a0:f6:fe:03:1e	Surfline4GLTE-031E	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:06	1	-94	5.5727456	-0.20269	46.37859	16	WIFI
14	6a:a0:f6:fd:dc:93	Surfline4GLTE-DC92	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 14:35	1	-96	5.56833003	-0.22038	32.23633	12	WIFI
15	6a:a0:f6:fd:e0:c5	Surfline4GLTE-E0C4	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:12	4	-94	5.56504123	-0.17728	36.43592	24	WIFI
16	6a:a0:f6:fd:ec:41	Surfline4GLTE-EC40	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:28	1	-99	5.56759593	-0.12776	38.50473	12	WIFI
17	6a:a0:f6:fd:f1:bb	Surfline4GLTE-F1BA	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 14:19	6	-94	5.53369479	-0.23768	47.145	12	WIFI
18	6a:a0:f6:ff:6a:86	Surfline4GLTE-6A85	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:35	6	-93	5.58476929	-0.10278	38.20089	16	WIFI
19	6a:a0:f6:ff:73:cc	Surfline4GLTE-73CB	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 14:30	1	-93	5.54888957	-0.22911	39.69072	16	WIFI
20	78:6a:89:43:56:63	WLAN1-460060507000042	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:10	11	-92	5.56746961	-0.1819	46.22948	24	WIFI
21	78:6a:89:43:57:71	WLAN1-460060507000096	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 14:32	11	-87	5.5610955	-0.22924	41.65688	8	WIFI
22	78:6a:89:43:57:f8	WLAN1-460060507000123	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 14:32	6	-95	5.55997252	-0.23008	41.72316	12	WIFI
23	78:6a:89:43:58:57	WLAN1-460060507000142	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:10	1	-95	5.5678547	-0.18263	44.01716	12	WIFI
24	78:6a:89:43:58:89	WLAN1-460060507000152	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:10	1	-96	5.56746961	-0.1819	46.22948	24	WIFI
25	78:6a:89:43:59:88	WLAN1-460060507000203	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 14:32	6	-93	5.56078783	-0.22957	43.40034	8	WIFI
26	78:6a:89:43:5a:0f	WLAN1-460060507000230	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:10	1	-96	5.56746961	-0.1819	46.22948	24	WIFI
27	78:6a:89:43:5a:2d	WLAN1-460060507000236	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:06	1	-89	5.5731412	-0.20088	33.02934	16	WIFI
28	78:6a:89:43:5a:eb	WLAN1-460060507000274	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:09	11	-80	5.56935671	-0.18525	48.4254	24	WIFI
29	78:6a:89:43:5b:0e	WLAN1-460060507000281	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:09	11	-75	5.56935671	-0.18525	48.4254	24	WIFI
30	84:a8:e4:1a:78:a0	skan network	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:47	1	-94	5.59981519	-0.0964	42.66542	6	WIFI
31	84:a8:e4:1a:9d:f8	Danny Computers	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 14:31	1	-88	5.55942456	-0.23039	35.84695	8	WIFI
32	84:a8:e4:1a:b7:90	mqfamily	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:41	1	-97	5.59275264	-0.0941	29.34976	12	WIFI
33	84:a8:e4:1a:d5:8c	Ghana Police	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:58	1	-99	5.59456339	-0.08787	44.53646	8	WIFI
34	90:f6:52:76:0d:0d	ALBRIM ACCESS	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:15	4	-89	5.55649571	-0.16651	38.80474	16	WIFI
35	bc:76:70:67:0d:c4	winsel	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:31	1	-96	5.58030958	-0.1078	40.39099	16	WIFI
36	c2:9f:db:83:ad:46	Sport2	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:04	3	-91	5.57113325	-0.20891	47.08502	12	WIFI
37	c8:d3:a3:2c:8b:52	BCD Wan	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:12	11	-87	5.56439272	-0.1762	31.2838	16	WIFI
38	c8:d5:fe:1e:39:18	LUFTHANSA ... NON STOP YOU	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:06	6	-95	5.57284662	-0.20223	45.28488	16	WIFI
39	cc:96:a0:da:29:88	B APPAH	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:12	1	-94	5.56321992	-0.17432	48.53928	24	WIFI
40	cc:b2:55:dd:5c:c8	SDTM	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 14:33	1	-94	5.56615658	-0.22448	40.58934	16	WIFI
41	ea:08:8b:73:3b:d7	Komla Klutse	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:23	1	-98	5.55837437	-0.16111	41.66029	12	WIFI
42	00:18:e7:de:1e:88	TATA ACCRA-OFFICE	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 14:33	6	-91	5.56335811	-0.22758	40.7988	12	WIFI
43	00:18:e7:fb:0d:a3	TNUNGUA-STO	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:37	11	-94	5.58761951	-0.09932	36.11609	16	WIFI
44	00:23:69:61:ac:ea	VIRUS	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 13:35	1	-88	5.53960542	-0.24348	-455.14	400	WIFI
45	00:25:9c:d9:74:1b	linksys	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:10	11	-93	5.56746961	-0.1819	46.22948	24	WIFI
46	08:7a:4c:51:7c:5c	MALBERNICE	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 14:27	11	-94	5.53659838	-0.22436	54.70074	12	WIFI
47	08:7a:4c:51:84:20	Fujifilm	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:05	1	-93	5.57212765	-0.20492	45.8138	16	WIFI
48	08:7a:4c:51:92:40	novotec	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:12	10	-93	5.56504123	-0.17728	36.43592	24	WIFI
49	08:7a:4c:51:9c:dc	FGBMFI	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:13	1	-91	5.5619315	-0.17211	46.96099	12	WIFI
50	08:7a:4c:51:9d:ec	KOFIKROM NET	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:23	1	-79	5.55818617	-0.16202	43.56668	16	WIFI
51	08:7a:4c:51:a0:6c	dejongltd	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:09	1	-88	5.56935671	-0.18525	48.4254	24	WIFI
52	08:7a:4c:51:a3:5c	millitechltd	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 14:30	3	-95	5.55528006	-0.23175	29.33292	12	WIFI
53	08:7a:4c:51:c9:f4	-----NEXTit-----	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:22	11	-94	5.55724135	-0.16445	47.71476	12	WIFI
54	08:7a:4c:52:18:ac	Greenland	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 14:27	10	-82	5.53644067	-0.22517	50.28102	24	WIFI
55	08:7a:4c:98:d3:dc	tenda	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:07	1	-93	5.57401659	-0.19733	37.26557	16	WIFI
56	08:7a:4c:98:de:84	kellykelly	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:54	2	-92	5.60094537	-0.09357	40.57494	8	WIFI
57	08:7a:4c:98:fa:40	Multivet (Gh) Ltd	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:15	1	-90	5.55605057	-0.16761	42.57668	12	WIFI

	A	B	C	D	E	F	G	H	I	J	K
1	WigleWifi-1.4	appRelease=2.7	model=HUAWEI MT7-L09	release=4.4.2	device=hwmt7	display=MT7-L09V100R001C00B126	board=MT7-L09	brand=Huawei			
2	MAC	SSID	AuthMode	FirstSeen	Channel	RSSI	CurrentLatitude	CurrentLongitude	AltitudeMeters	AccuracyMeters	Type
3	08:7a:4c:99:09:b0	prince	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:45	1	-96	5.59806899	-0.09625	39.23457	12	WIFI
4	08:7a:4c:99:33:f0	BUASEA NET	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:22	11	-83	5.55772527	-0.16322	50.03623	12	WIFI
5	08:7a:4c:99:50:04	I-ZAR	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:05	11	-89	5.57247007	-0.20366	40.87886	16	WIFI
6	08:7a:4c:99:b5:54	MFMTESHIE	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:54	1	-94	5.60047168	-0.09466	37.31145	8	WIFI
7	08:7a:4c:99:bf:c8	MedexPharmacy	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:11	1	-96	5.56635778	-0.17979	49.23404	12	WIFI
8	08:7a:4c:99:e4:74	Dukewilliams	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 13:38	1	-97	5.5363857	-0.2439	64.17019	32	WIFI
9	08:7a:4c:99:ea:3c	INVISIBLE2014	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 14:31	1	-90	5.55735558	-0.23154	28.27366	12	WIFI
10	08:7a:4c:9a:28:4c	KASANTE	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 14:30	3	-92	5.55528006	-0.23175	29.33292	12	WIFI
11	08:7a:4c:9a:43:9c	EDDYSPIZZA	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:05	11	-88	5.57181212	-0.20634	44.51825	16	WIFI
12	08:7a:4c:9a:64:38	C.A DIVINE GSM	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:03	1	-95	5.56990464	-0.21403	42.45295	24	WIFI
13	08:7a:4c:9a:8c:88	ST MARY'S	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 14:27	1	-91	5.53678298	-0.22352	51.84065	12	WIFI
14	08:7a:4c:9a:99:fc	Ghanahrsolutions	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:22	2	-93	5.55724135	-0.16445	47.71476	12	WIFI
15	0a:86:3b:0b:23:a2	odameguest	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:08	11	-92	5.57093199	-0.18812	47.73661	12	WIFI
16	14:b9:68:42:9f:d9	MALMSURFLINE	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:11	6	-92	5.56681957	-0.18101	49.64999	12	WIFI
17	14:b9:68:42:a0:19	SURFLINE4G-A019	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:04	6	-87	5.57098183	-0.20945	45.15221	16	WIFI
18	14:b9:68:42:a5:03	SURFLINE4G-A503	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:15	6	-89	5.55661005	-0.16619	43.02736	24	WIFI
19	28:10:7b:f7:37:d4	MMWireless	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:29	8	-86	5.57582056	-0.11128	46.86737	12	WIFI
20	48:f8:b3:a5:d0:86	AMISGOLD MICROFINANCE	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:04	3	-93	5.57033497	-0.21167	25.9256	24	WIFI
21	58:6d:8f:9f:78:09	ROCKPHARM	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 14:26	6	-92	5.5360753	-0.22683	50.68963	16	WIFI
22	5c:d9:98:63:2d:a8	MS-dlink	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:06	5	-84	5.57260096	-0.20318	45.603	16	WIFI
23	68:a0:f6:99:0d:de	MAX-WIFI 4G	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:22	1	-93	5.55680504	-0.16556	39.67743	12	WIFI
24	68:a0:f6:99:0e:3c	SURFLINE4G-0E3C	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:37	1	-92	5.58677781	-0.10048	39.67097	12	WIFI
25	68:a0:f6:99:0f:e8	SURFLINE4G-0FE8	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:54	6	-88	5.60001621	-0.0958	42.24795	8	WIFI
26	6c:e8:73:3b:07:9c	TP-LINK_3B079C	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:07	6	-94	5.5742468	-0.1962	35.15703	16	WIFI
27	70:72:3c:99:d7:4f	Airtel-D74D	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:16	11	-90	5.55670023	-0.1657	42.98417	16	WIFI
28	84:db:ac:c7:c3:7c	m-solutions-adsl	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:07	11	-89	5.57413391	-0.19675	38.20125	16	WIFI
29	84:db:ac:c8:33:0c	FAMILYHEALTHHOSPITAL.NET	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:29	3	-95	5.57322698	-0.11346	36.79532	16	WIFI
30	84:db:ac:c8:38:7c	Little Fish	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:56	2	-83	5.59631211	-0.0896	50.72504	12	WIFI
31	84:db:ac:c8:45:a4	Rephealth	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 14:23	1	-92	5.53730813	-0.22895	41.16402	12	WIFI
32	84:db:ac:c8:9d:64	KUNG FU FISH	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:15	11	-74	5.55661005	-0.16619	43.02736	24	WIFI
33	84:db:ac:c8:cc:a8	ISTC TRAINING SCHOOL	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 14:33	1	-92	5.56542289	-0.22528	41.32834	16	WIFI
34	98:fc:11:45:c7:4a	kofi_blinx	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:54	6	-88	5.60094537	-0.09357	40.57494	8	WIFI
35	9c:d6:43:c9:6b:9e	dlink	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:09	2	-91	5.5695797	-0.18566	44.94277	16	WIFI
36	a4:99:47:31:7a:0c	LE MAGELLAN	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:12	1	-95	5.56273487	-0.17351	38.81092	16	WIFI
37	a4:99:47:31:ba:24	MEDBOOKS LTD	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 14:22	11	-74	5.53508611	-0.23152	49.06529	16	WIFI
38	a4:99:47:31:c9:e0	VICDORIS2	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:05	9	-87	5.57200929	-0.20566	48.88391	16	WIFI
39	a4:99:47:3f:bb:e4	legalconsult	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:15	11	-87	5.55563343	-0.16887	45.01266	12	WIFI
40	a4:99:47:3f:c3:1c	JOKERCARD CASINO	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:38	11	-96	5.58809619	-0.09842	37.63987	8	WIFI
41	a4:99:47:3f:da:5c	VDF-HG532e-KEYQD4	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 14:33	10	-85	5.56400403	-0.22682	40.64447	12	WIFI
42	a4:99:47:40:2a:c0	PRIS-B TRAVEL AND TOUR	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:57	1	-97	5.59411499	-0.08874	39.35269	12	WIFI
43	a4:99:47:40:2c:88	NANA -CBD -NET	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:12	1	-86	5.56439272	-0.1762	31.2838	16	WIFI
44	a4:99:47:40:43:b0	R-NET	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:25	1	-97	5.55999191	-0.14843	39.53531	16	WIFI
45	a4:99:47:40:43:d8	advent_press	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:14	1	-88	5.55615794	-0.16785	46.10223	16	WIFI
46	a4:99:47:40:65:5c	GempSEC	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 14:23	11	-88	5.5373009	-0.2291	40.72928	24	WIFI
47	a4:99:47:40:83:28	kwekuAntwi	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 14:27	2	-95	5.53613401	-0.22653	49.4381	16	WIFI
48	a4:99:47:40:90:a0	SPRINGBOARD	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 14:21	10	-78	5.53474895	-0.23312	45.52926	16	WIFI
49	a4:99:47:40:b4:fc	robertandsonsadmin	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:12	10	-94	5.56564655	-0.17842	47.80924	16	WIFI
50	a4:99:47:40:d6:30	martha	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:04	1	-94	5.57020317	-0.21274	37.95327	12	WIFI
51	a4:99:47:40:d7:b8	JPITCAMPUS	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:04	2	-91	5.57033497	-0.21167	25.9256	24	WIFI
52	ac:f1:df:22:a3:f0	M&J	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:10	3	-87	5.5687149	-0.18415	49.38645	12	WIFI
53	ac:f1:df:c8:1b:e2	Expat	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:12	1	-95	5.56273487	-0.17351	38.81092	16	WIFI
54	b0:48:7a:c8:8d:02	FNSL1	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:06	4	-87	5.57311573	-0.20107	30.32192	16	WIFI
55	c0:a0:bb:c5:33:82	Evolution1	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:09	3	-94	5.57008869	-0.18658	47.33652	16	WIFI
56	c8:3a:35:0e:e4:e8	HomeNet	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:14	6	-80	5.55615794	-0.16785	46.10223	16	WIFI
57	c8:3a:35:1d:4f:e8	Media Net	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:05	6	-93	5.57247007	-0.20366	40.87886	16	WIFI

	A	B	C	D	E	F	G	H	I	J	K
1	WigleWifi-1.4	appRelease=2.7	model=HUAWEI MT7-L09	release=4.4.2	device=hwmt7	display=MT7-L09V100R001C00B126	board=MT7-L09	brand=Huawei			
2	MAC	SSID	AuthMode	FirstSeen	Channel	RSSI	CurrentLatitude	CurrentLongitude	AltitudeMeters	AccuracyMeters	Type
3	c8:d7:19:ff:90:92	Provident-life	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:04	1	-87	5.5703827	-0.21118	30.84723	16	WIFI
4	c8:d7:19:ff:94:d4	HRGWorldWide	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:12	11	-89	5.56374764	-0.17518	46.02924	16	WIFI
5	cc:b2:55:d4:20:fa	CASH POINT	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:36	1	-98	5.58655922	-0.10072	38.47755	16	WIFI
6	d4:6e:5c:ec:18:5c	EGL-Voda2	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:04	4	-83	5.57113325	-0.20891	47.08502	12	WIFI
7	d4:6e:5c:ec:27:f0	OMNITRUST	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:58	1	-84	5.59512835	-0.08691	45.66571	8	WIFI
8	d4:6e:5c:ec:3d:6c	caesars vodafone	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:11	1	-93	5.56628824	-0.17968	49.15025	12	WIFI
9	d4:6e:5c:ec:5a:e0	ADMINISTRATOR	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:07	1	-92	5.5739089	-0.19356	29.17934	16	WIFI
10	d4:6e:5c:ec:6e:b0	winterdreams	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:22	1	-92	5.55689428	-0.16533	43.11257	8	WIFI
11	d4:6e:5c:ec:d2:1c	GLORYGATE CAPITAL	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:37	1	-96	5.58723612	-0.09993	38.30208	16	WIFI
12	d4:6e:5c:ec:d8:34	ericdai net	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:58	1	-95	5.59487029	-0.08737	46.92725	8	WIFI
13	d4:6e:5c:ec:de:9c	THY WILL	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 14:22	2	-92	5.53513447	-0.23131	48.76327	16	WIFI
14	e8:94:f6:6b:48:86	UDS GUEST HOUSE	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:08	1	-94	5.57168272	-0.18966	47.46498	16	WIFI
15	f0:84:c9:d2:1c:13	Glo Mobile WiFi_D21C13	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:06	11	-80	5.5733465	-0.20012	34.69948	16	WIFI
16	f4:ec:38:d0:0d:90	DELCIELO	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:06	3	-84	5.5727456	-0.20269	46.37859	16	WIFI
17	f4:ec:38:dd:ad:c6	EMMA CAFE	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:32	4	-90	5.58318659	-0.10452	46.62997	32	WIFI
18	fc:75:16:e7:5e:b8	Sammy	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:09	1	-93	5.5702644	-0.18694	50.49103	16	WIFI
19	00:1d:0f:e6:2f:1a	CASCAD MEDICAL SUPPLIES	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP-preauth][ESS]	2/10/2015 14:21	6	-87	5.5348677	-0.23256	47.14416	16	WIFI
20	00:23:cd:d4:41:c2	Baha'i Centre	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP-preauth][ESS]	2/10/2015 15:05	6	-94	5.57247007	-0.20366	40.87886	16	WIFI
21	00:23:cd:f9:12:5d	FLP_Admin2	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP-preauth][ESS]	2/10/2015 15:02	6	-87	5.56927347	-0.21562	35.95197	8	WIFI
22	00:25:86:c3:8f:60	TP-LINK_C38F60	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP-preauth][ESS]	2/10/2015 15:54	6	-93	5.60154021	-0.09199	46.20072	8	WIFI
23	54:e6:fc:c8:bd:b4	OPS-CID	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP-preauth][ESS]	2/10/2015 15:09	5	-90	5.5698079	-0.18605	44.56901	12	WIFI
24	a0:f3:c1:ba:48:30	WFS AP	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP-preauth][ESS]	2/10/2015 14:20	10	-71	5.53392094	-0.23687	41.24313	8	WIFI
25	b0:48:7a:99:5b:4c	PayAsYouGo 2	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP-preauth][ESS]	2/10/2015 15:09	11	-92	5.5698079	-0.18605	44.56901	12	WIFI
26	1c:7e:e5:34:82:e4	Ebenos	[WPA-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 14:21	5	-89	5.53432321	-0.23502	44.51236	6	WIFI
27	5c:d9:98:63:2d:74	ServerTeam	[WPA-PSK-CCMP+TKIP][WPS][ESS]	2/10/2015 15:06	7	-83	5.57260096	-0.20318	45.603	16	WIFI
28	00:15:6d:72:43:03	PUCITY-B	[WPA-PSK-TKIP][ESS]	2/10/2015 15:13	4	-85	5.56119482	-0.17043	58.13226	24	WIFI
29	00:16:b6:bd:ed:ae	SOLAR LAND	[WPA-PSK-TKIP][ESS]	2/10/2015 15:22	6	-82	5.55772527	-0.16322	50.03623	12	WIFI
30	00:18:f8:a2:28:03	FC	[WPA-PSK-TKIP][ESS]	2/10/2015 15:07	6	-95	5.5742468	-0.1962	35.15703	16	WIFI
31	00:1d:2e:30:5d:e1		[WPA-PSK-TKIP][ESS]	2/10/2015 15:04	6	-78	5.5703827	-0.21118	30.84723	16	WIFI
32	00:1d:2e:70:5d:e1	PIC-AP-01	[WPA-PSK-TKIP][ESS]	2/10/2015 15:04	6	-79	5.5703827	-0.21118	30.84723	16	WIFI
33	00:22:3f:7d:a5:54	NETGEAR	[WPA-PSK-TKIP][ESS]	2/10/2015 15:04	6	-90	5.57113325	-0.20891	47.08502	12	WIFI
34	00:27:22:76:64:41	KAIPTC-WIFI M	[WPA-PSK-TKIP][ESS]	2/10/2015 15:29	6	-92	5.57322698	-0.11346	36.79532	16	WIFI
35	84:a8:e4:1a:f3:bc	Test	[WPA-PSK-TKIP][ESS]	2/10/2015 15:22	1	-94	5.55697142	-0.16511	46.75422	12	WIFI
36	84:a8:e4:1b:3a:50	RockChemists	[WPA-PSK-TKIP][ESS]	2/10/2015 14:27	1	-94	5.53613401	-0.22653	49.4381	16	WIFI
37	bc:76:70:66:e6:2c	Benco Wifi	[WPA-PSK-TKIP][ESS]	2/10/2015 14:21	1	-88	5.53463578	-0.23369	45.7006	16	WIFI
38	bc:76:70:67:03:e4	CAPITAL&MORE	[WPA-PSK-TKIP][ESS]	2/10/2015 15:04	1	-96	5.57141072	-0.20787	45.91128	16	WIFI
39	bc:76:70:67:03:e4	CAPITAL&MORE	[WPA-PSK-TKIP][ESS]	2/10/2015 15:05	1	-96	5.57200929	-0.20566	48.88391	16	WIFI
40	c0:c1:c0:1d:b8:18	EURO(ZEPNET)	[WPA-PSK-TKIP][ESS]	2/10/2015 15:06	11	-81	5.57304565	-0.20142	36.14104	16	WIFI
41	c8:3a:35:f2:3f:68	WFS2	[WPA-PSK-TKIP][ESS]	2/10/2015 14:20	6	-91	5.53392094	-0.23687	41.24313	8	WIFI
42	cc:96:a0:ce:a2:77	NISA COMPUTERS	[WPA-PSK-TKIP][ESS]	2/10/2015 15:14	6	-93	5.55592355	-0.16848	48.49606	16	WIFI
43	cc:96:a0:da:36:e0	GABON	[WPA-PSK-TKIP][ESS]	2/10/2015 14:32	6	-94	5.56194109	-0.22888	40.55934	16	WIFI
44	dc:9f:db:0c:1c:48	KAIPTC-WIFI 1	[WPA-PSK-TKIP][ESS]	2/10/2015 15:29	1	-92	5.57408126	-0.11244	44.00741	16	WIFI
45	5c:96:9d:64:9e:4f		[WPA-PSK-TKIP][WPA2-PSK-CCMP+TKIP][ESS]	2/10/2015 15:05	1	-91	5.57194746	-0.20593	45.92768	24	WIFI
46	68:7f:74:e7:d8:70	Acumed	[WPA-PSK-TKIP][WPA2-PSK-CCMP-preauth][WPS][ESS]	2/10/2015 14:26	9	-92	5.53596384	-0.22732	51.65093	16	WIFI
47	98:fc:11:bf:ba:1a	BROADVIEW	[WPA-PSK-TKIP][WPA2-PSK-CCMP-preauth][WPS][ESS]	2/10/2015 15:06	6	-93	5.5727456	-0.20269	46.37859	16	WIFI
48	00:e0:4d:70:45:58	magnalink	[WPA-PSK-TKIP][WPA2-PSK-TKIP][ESS]	2/10/2015 15:06	6	-91	5.57260096	-0.20318	45.603	16	WIFI
49	84:a8:e4:1a:96:90	spyglover	[WPA-PSK-TKIP][WPA2-PSK-TKIP][ESS]	2/10/2015 14:18	1	-87	5.53549199	-0.24268	49.5736	12	WIFI
50	84:a8:e4:1a:c1:f8	GHANA SYTO	[WPA-PSK-TKIP][WPA2-PSK-TKIP][ESS]	2/10/2015 15:10	6	-91	5.5687149	-0.18415	49.38645	12	WIFI
51	84:a8:e4:1a:ce:a4	Dzotronics Systems	[WPA-PSK-TKIP][WPA2-PSK-TKIP][ESS]	2/10/2015 14:22	1	-89	5.53525072	-0.23077	38.48704	12	WIFI
52	84:a8:e4:1a:e3:50		[WPA-PSK-TKIP][WPA2-PSK-TKIP][ESS]	2/10/2015 14:32	6	-96	5.56145257	-0.22905	36.60393	12	WIFI
53	bc:76:70:66:b5:94	ANKAMAH AND ASSOC	[WPA-PSK-TKIP][WPA2-PSK-TKIP][ESS]	2/10/2015 15:06	4	-92	5.5727456	-0.20269	46.37859	16	WIFI
54	cc:96:a0:ce:98:eb	JAZZ 1	[WPA-PSK-TKIP][WPA2-PSK-TKIP][ESS]	2/10/2015 15:07	1	-92	5.57430869	-0.1946	34.77998	16	WIFI
55	cc:96:a0:da:02:84	Irisnetwork	[WPA-PSK-TKIP][WPA2-PSK-TKIP][ESS]	2/10/2015 15:54	1	-94	5.60141519	-0.09233	45.43724	8	WIFI
56	cc:96:a0:da:22:80	KIDNEYRESEARCH1	[WPA-PSK-TKIP][WPA2-PSK-TKIP][ESS]	2/10/2015 14:23	11	-88	5.53706721	-0.22923	46.33804	16	WIFI
57	00:22:b0:f4:0e:31	dlink	[WPA-PSK-TKIP][WPA2-PSK-TKIP][WPS][ESS]	2/10/2015 14:30	6	-93	5.55289685	-0.23091	40.07037	12	WIFI

	A	B	C	D	E	F	G	H	I	J	K
1	WigleWifi-1.4	appRelease=2.7	model=HUAWEI MT7-L09	release=4.4.2	device=hw mt7	display=MT7-L09V100R001C00B126	board=MT7-L09	brand=Huawei			
2	MAC	SSID	AuthMode	FirstSeen	Channel	RSSI	CurrentLatitude	CurrentLongitude	AltitudeMeters	AccuracyMeters	Type
3	00:21:29:67:46:5d	Leo	[WPA-PSK-TKIP][WPS][ESS]	2/10/2015 15:02	6	-91	5.56921293	-0.21534	32.22188	16	WIFI
4	00:22:6b:74:94:47	DE WIRELESS	[WPA-PSK-TKIP][WPS][ESS]	2/10/2015 15:05	3	-87	5.57247007	-0.20366	40.87886	16	WIFI
5	00:22:6b:8e:b5:e1	chinamedteam	[WPA-PSK-TKIP][WPS][ESS]	2/10/2015 14:30	1	-96	5.54832238	-0.22881	42.57436	16	WIFI
6	34:cd:be:5f:8c:26	B660-8C24	[WPA-PSK-TKIP][WPS][ESS]	2/10/2015 14:33	6	-93	5.56579833	-0.22486	38.6822	12	WIFI
7	74:ea:3a:db:a6:58	Mugabe	[WPS][ESS]	2/10/2015 15:04	6	-93	5.57113325	-0.20891	47.08502	12	WIFI
8	d8:eb:97:a1:ab:9c	ECO-TWO	[WPS][ESS]	2/10/2015 15:09	1	-94	5.5695797	-0.18566	44.94277	16	WIFI
9	e8:94:f6:6b:11:02	TP-LINK_6B1102	[WPS][ESS]	2/10/2015 15:11	1	-94	5.56718225	-0.18135	50.57257	24	WIFI
10	e8:de:27:67:73:68	kodako	[WPS][ESS]	2/10/2015 15:13	6	-90	5.56251209	-0.17313	43.37319	16	WIFI
11	f8:d1:11:48:07:32	TP-LINK_480732	[WPS][ESS]	2/10/2015 15:34	1	-95	5.58416302	-0.10348	39.20425	16	WIFI
12	00:18:e7:f9:0a:1c	vertex	[WPS][WEP][ESS]	2/10/2015 15:58	5	-81	5.59501186	-0.08712	46.50939	6	WIFI
13	00:25:9c:67:51:95	Jubilee(Ucom)	[WPS][WEP][ESS]	2/10/2015 15:09	8	-96	5.56935671	-0.18525	48.4254	24	WIFI
14	00:26:5a:c4:60:37	ECL	[WPS][WEP][ESS]	2/10/2015 15:04	6	-93	5.57020317	-0.21274	37.95327	12	WIFI
15	00:26:5a:c4:60:6f	ECL	[WPS][WEP][ESS]	2/10/2015 15:04	6	-94	5.57018204	-0.21289	37.27497	32	WIFI
16	64:70:02:84:fe:be	Vizico	[WPS][WEP][ESS]	2/10/2015 15:39	3	-96	5.58859937	-0.09765	42.49541	12	WIFI
17	c0:c1:c0:f4:9c:b4	AMBO(ZipNeT)	[WPS][WEP][ESS]	2/10/2015 15:06	3	-91	5.5727456	-0.20269	46.37859	16	WIFI
18	f4:ec:38:dd:a2:a4	BLACKS SECRET	[WPS][WEP][ESS]	2/10/2015 15:21	4	-94	5.55669152	-0.16568	38.73942	16	WIFI
19	f8:d1:11:8d:8f:3a	ShawbellC	[WPS][WEP][ESS]	2/10/2015 15:12	3	-92	5.56374764	-0.17518	46.02924	16	WIFI

Microsoft Excel - WigleWifi_20130403203507

	A	B	C	D	E	F	G	H	I	J	K
1	WigleWifi-1.4	appRelease=1.55	model=HTC Wildfire S A510e	release=2.3.5	device=ma	display=G	board=marv	brand=htc_europe			
2	MAC	SSID	AuthMode	FirstSeen	Channe	RSSI	CurrentLatitude	CurrentLongitude	Altitude Meters	Accuracy Meters	Type
3	00:1a:2f:7f:c8:50	Achimota_branch	[WEP]	2/16/2013 11:00	11	-92	5.624131	-0.232918	59	6	WIFI
4	00:15:e9:ba:fa:54	alabee	[WEP]	1/19/2013 13:22	6	-90	5.6431961	-0.112905	69	6	WIFI
5	00:21:29:89:aa:90	Allurespainthecity	[WEP]	2/26/2013 11:35	11	-94	5.5585724	-0.167917	45	4	WIFI
6	00:1f:1f:db:0c:58	Anastasia's Boutique	[WEP]	1/19/2013 13:25	11	-83	5.6356323	-0.1316	84	3	WIFI
7	00:21:29:6a:0a:49	Dez_Amis	[WEP]	2/26/2013 12:27	1	-94	5.5753362	-0.188441	74	3	WIFI
8	bc:76:70:66:d2:bc	EAKAZA_LOBBY	[WEP]	1/18/2013 8:22	1	-90	5.6355143	-0.13167	79	32	WIFI
9	58:8d:09:e2:d2:4c	Family health Hospital	[WEP]	2/26/2013 9:38	1	-86	5.5735981	-0.113307	41	2	WIFI
10	00:e0:4d:cf:b0:36	HG520c	[WEP]	4/3/2013 20:10	1	-94	5.5543613	-0.179375	32	12	WIFI
11	00:e0:4d:cf:d6:22	HG520c	[WEP]	3/21/2013 11:26	1	-92	5.5904049	-0.181103	80	4	WIFI
12	84:a8:e4:1a:9e:f4	HG520c	[WEP]	1/1/1970 0:00	6	-88	5.5459231	-0.202582	0	310.165	WIFI
13	84:a8:e4:1a:ba:b8	HG520c	[WEP]	1/29/2013 10:29	1	-84	5.6334275	-0.13712	71	6	WIFI
14	84:a8:e4:1a:be:e0	HG520c	[WEP]	2/26/2013 11:43	6	-90	5.5709374	-0.188543	63	2	WIFI
15	84:a8:e4:1a:d0:54	HG520c	[WEP]	1/29/2013 10:10	1	-87	5.6269795	-0.085348	59	3	WIFI
16	84:a8:e4:1a:d0:74	HG520c	[WEP]	1/19/2013 13:17	11	-86	5.6342161	-0.098705	68	4	WIFI
17	84:a8:e4:1a:d3:94	HG520c	[WEP]	2/16/2013 10:58	1	-92	5.6094003	-0.226942	48	4	WIFI
18	84:a8:e4:1a:d8:2c	HG520c	[WEP]	2/26/2013 9:35	1	-93	5.5778199	-0.110164	34	6	WIFI
19	84:a8:e4:1b:21:cc	HG520c	[WEP]	2/26/2013 12:32	1	-86	5.5861938	-0.183882	81	3	WIFI
20	cc:96:a0:da:2e:f8	HG520c	[WEP]	2/16/2013 9:29	7	-85	5.6381106	-0.125501	77	4	WIFI
21	00:1f:1f:ca:df:a5	Hotline	[WEP]	2/16/2013 9:31	11	-90	5.6352943	-0.132458	84	3	WIFI
22	34:08:04:d8:60:00	Ikiki Show room	[WEP]	2/16/2013 9:31	7	-81	5.6354338	-0.132179	88	3	WIFI
23	00:15:6d:4c:a7:a2	iTrack	[WEP]	1/29/2013 10:27	6	-78	5.6356913	-0.13123	86	8	WIFI
24	00:27:22:8a:f9:15	LaVillawifi_1	[WEP]	2/26/2013 11:39	1	-90	5.567928	-0.182514	49	2	WIFI
25	00:0c:41:79:9a:50	linksys	[WEP]	4/2/2013 13:56	6	-88	5.5625689	-0.181623	47	4	WIFI
26	00:26:5a:c4:5c:83	MaxMartOffice	[WEP]	2/26/2013 12:37	1	-94	5.5917084	-0.180679	80	4	WIFI
27	00:09:5b:67:7b:59	NETGEAR	[WEP]	1/1/1970 0:00	11	-84	5.6647825	-0.083462	0	283.53	WIFI
28	00:13:10:f2:18:d6	nexus	[WEP]	1/19/2013 13:13	6	-93	5.6273389	-0.086029	60	4	WIFI
29	1c:7e:e5:d3:60:ba	Officepal	[WEP]	2/26/2013 13:06	6	-91	5.63052	-0.143611	86	12	WIFI
30	90:f6:52:64:2d:e8	OHAYO-Y.T	[WEP]	2/26/2013 12:49	9	-96	5.6176615	-0.175921	76	6	WIFI
31	00:60:b3:2c:6a:3d	Penta_WiFi_Zone1	[WEP]	4/2/2013 13:51	3	-89	5.5652833	-0.181172	54	6	WIFI
32	00:60:b3:07:1d:de	Penta_WiFi_Zone2	[WEP]	4/2/2013 13:51	11	-96	5.5651921	-0.181199	54	3	WIFI
33	00:25:86:d9:76:f8	PVet	[WEP]	1/29/2013 10:12	6	-90	5.628165	-0.088336	65	2	WIFI
34	00:27:19:dc:f0:74	PWPLANT1	[WEP]	2/26/2013 13:02	3	-93	5.6261319	-0.157022	84	4	WIFI
35	90:f6:52:f0:5d:cc	RGEMC	[WEP]	3/21/2013 13:16	3	-94	5.6297261	-0.141197	61	2	WIFI
36	00:1c:f0:9d:95:13	Robin 1st Floor	[WEP]	1/1/1970 0:00	1	-90	5.6507767	-0.099375	0	620.764	WIFI

3	00:1c:f0:9d:94:e6	Robin 2nd Floor	[WEP]	1/19/2013 13:27	1	-89	5.6323278	-0.139765	72	6	WIFI
4	00:1c:f0:9d:95:16	Robin 3rd Floor	[WEP]	3/21/2013 13:17	1	-88	5.6303161	-0.140451	56	2	WIFI
5	00:1c:f0:9d:95:1b	Robin 4th Floor	[WEP]	3/21/2013 13:17	1	-93	5.6302571	-0.140499	56	2	WIFI
6	00:1c:f0:9d:95:d7	Robin 5th Floor	[WEP]	1/18/2013 8:26	1	-90	5.6324619	-0.139625	72	8	WIFI
7	00:23:04:2f:4b:c0	Sakumono_branch	[WEP]	1/19/2013 13:06	2	-94	5.6244743	-0.074539	73	3	WIFI
8	00:1d:0f:ea:ec:c4	Shawbell	[WEP]	2/26/2013 11:37	6	-93	5.5639637	-0.175384	45	3	WIFI
9	00:25:86:b4:97:fe	Shawbell	[WEP]	2/26/2013 11:37	6	-90	5.5639637	-0.175384	45	3	WIFI

	A	B	C	D	E	F	G	H	I	J	K
1	WigleWifi-1.4	appRelease=1.55	model=HTC Wildfire S A510e	release=2.3.5	device=ma	display=G	board=marv	brand=htc_europe			
2	MAC	SSID	AuthMode	FirstSeen	Channe	RSSI	CurrentLatitude	CurrentLongitude	AltitudeMeters	AccuracyMeters	Type
3	10:9a:dd:ba:b6:7b	SIC-TFC	[WEP]	2/26/2013 11:39	11	-96	5.5686682	-0.183882	52	3	WIFI
4	00:12:0e:2d:b0:7c	speed	[WEP]	1/29/2013 10:10	6	-96	5.6271243	-0.08575	59	6	WIFI
5	00:3a:99:eb:fa:e0	Spintex_Warehouse	[WEP]	2/26/2013 13:02	2	-95	5.626722	-0.154286	87	8	WIFI
6	00:13:46:4c:5e:2e	SSAL	[WEP]	3/21/2013 11:27	6	-82	5.5994117	-0.178978	82	4	WIFI
7	84:a8:e4:1b:22:58	TAWIAH P	[WEP]	2/26/2013 12:35	1	-91	5.5879426	-0.182599	77	6	WIFI
8	00:13:5e:51:25:9c	TechMGhana	[WEP]	1/29/2013 10:26	6	-92	5.6365979	-0.12902	84	6	WIFI
9	00:21:29:aa:ee:ad	The COP_GS_Res	[WEP]	4/3/2013 20:12	11	-99	5.5547154	-0.174703	40	48	WIFI
10	00:21:27:ec:1b:06	TP-LINK	[WEP]	3/21/2013 11:31	6	-86	5.6173235	-0.175958	82	4	WIFI
11	00:23:cd:d4:44:1a	TP-LINK	[WEP]	1/19/2013 13:14	6	-95	5.6276661	-0.086828	59	3	WIFI
12	00:25:86:d9:8b:72	TP-LINK_D98B72	[WEP]	1/29/2013 10:11	6	-77	5.6273979	-0.086432	62	3	WIFI
13	0c:d9:96:22:71:30	VFGHTP	[WEP]	1/1/1970 0:00	1	-52	5.6301825	-0.144382	0	22.6146	WIFI
14	0c:d9:96:39:ed:20	VFGHTP	[WEP]	1/1/1970 0:00	6	-49	5.6301884	-0.144493	0	22.6146	WIFI
15	0c:d9:96:39:f1:b0	VFGHTP	[WEP]	1/19/2013 13:32	6	-81	5.6300157	-0.143391	73	3	WIFI
16	0c:d9:96:39:f2:c0	VFGHTP	[WEP]	1/1/1970 0:00	6	-60	5.629882	-0.14355	0	32.9197	WIFI
17	0c:d9:96:39:fb:40	VFGHTP	[WEP]	1/1/1970 0:00	6	-72	5.6298876	-0.143547	0	32.9197	WIFI
18	0c:d9:96:4c:cd:00	VFGHTP	[WEP]	2/16/2013 9:39	1	-68	5.6307453	-0.143455	73	12	WIFI
19	18:33:9d:3e:db:10	VFGHTP	[WEP]	1/1/1970 0:00	1	-52	5.629882	-0.14355	0	32.9197	WIFI
20	64:d8:14:b3:dd:e0	VFGHTP	[WEP]	1/1/1970 0:00	1	-81	5.630048	-0.143462	0	32.9197	WIFI
21	64:d8:14:ee:65:60	VFGHTP	[WEP]	1/18/2013 8:28	6	-71	5.6307077	-0.143718	77	3	WIFI
22	64:d8:14:ee:74:d0	VFGHTP	[WEP]	1/1/1970 0:00	1	-61	5.6299976	-0.143489	0	32.9197	WIFI
23	b0:48:7a:a5:f9:5f	virus	[WEP]	4/2/2013 13:56	6	-92	5.5622578	-0.181671	50	2	WIFI
24	00:27:19:cb:20:da	Vizico	[WEP]	2/26/2013 9:25	1	-93	5.5887902	-0.097391	35	8	WIFI
25	bc:76:70:67:39:c4	Vodafone HG530 Home Gateway	[WEP]	2/16/2013 9:32	1	-94	5.6348062	-0.133703	81	4	WIFI
26	bc:76:70:67:51:d8	Vodafone HG530 Home Gateway	[WEP]	2/26/2013 9:24	1	-94	5.5893749	-0.096361	33	6	WIFI
27	00:1d:0f:e7:05:90	YvonneEx	[WEP]	1/19/2013 13:11	6	-89	5.6258798	-0.081174	59	2	WIFI
28	00:1f:6d:b8:5f:71	ZNGH	[WEP]	1/18/2013 8:21	1	-87	5.6365067	-0.129433	78	12	WIFI
29	00:1f:6d:bb:88:b1	ZNGH	[WEP]	1/1/1970 0:00	1	-82	5.637762	-0.126654	0	387.425	WIFI
30	00:1f:6d:bb:8f:61	ZNGH	[WEP]	1/29/2013 10:27	1	-91	5.6363779	-0.129535	83	8	WIFI
31	1c:aa:07:6e:08:70	AT-Secure	[WPA2-EAP-CCMP]	2/26/2013 11:43	11	-91	5.570991	-0.188903	64	2	WIFI
32	58:35:d9:d5:d1:20	AT-Secure	[WPA2-EAP-CCMP]	2/26/2013 11:43	1	-89	5.5710715	-0.188785	66	2	WIFI
33	e8:40:40:78:c5:a0	AT-Secure	[WPA2-EAP-CCMP]	2/26/2013 11:46	1	-93	5.5713719	-0.189353	64	2	WIFI
34	e8:40:40:ac:8d:90	AT-Secure	[WPA2-EAP-CCMP]	2/26/2013 11:43	11	-90	5.5710661	-0.188822	66	2	WIFI
35	e8:40:40:ac:e9:00	AT-Secure	[WPA2-EAP-CCMP]	2/26/2013 11:43	1	-88	5.5710232	-0.188844	66	2	WIFI
36	e8:40:40:ad:0d:90	AT-Secure	[WPA2-EAP-CCMP]	2/26/2013 11:44	6	-90	5.5708677	-0.188978	64	2	WIFI
3	00:24:a8:a1:6c:f1	EWA@ECN	[WPA2-EAP-CCMP]	2/16/2013 10:46	1	-95	5.6248283	-0.175819	70	4	WIFI
4	10:8c:cf:ea:cd:d0	OFFSHORE	[WPA2-EAP-CCMP]	3/21/2013 11:27	9	-95	5.6005061	-0.178689	84	6	WIFI
5	0c:d9:96:22:71:32	VF-Corporate	[WPA2-EAP-CCMP]	1/1/1970 0:00	1	-47	5.6301825	-0.144382	0	22.6146	WIFI
6	0c:d9:96:39:ed:22	VF-Corporate	[WPA2-EAP-CCMP]	1/1/1970 0:00	6	-58	5.6301854	-0.144437	0	22.6146	WIFI
7	0c:d9:96:39:f1:b2	VF-Corporate	[WPA2-EAP-CCMP]	1/29/2013 10:35	6	-78	5.6301123	-0.143412	68	6	WIFI
8	0c:d9:96:39:f2:c2	VF-Corporate	[WPA2-EAP-CCMP]	1/1/1970 0:00	6	-62	5.6301884	-0.144493	0	22.6146	WIFI
9	0c:d9:96:39:fb:42	VF-Corporate	[WPA2-EAP-CCMP]	2/16/2013 9:39	6	-84	5.6307453	-0.143455	73	12	WIFI
10	0c:d9:96:4c:cd:02	VF-Corporate	[WPA2-EAP-CCMP]	2/16/2013 9:39	1	-69	5.6307453	-0.143455	73	12	WIFI
11	18:33:9d:3e:db:12	VF-Corporate	[WPA2-EAP-CCMP]	1/1/1970 0:00	1	-72	5.6301708	-0.144159	0	22.6146	WIFI
12	64:d8:14:b3:dd:e2	VF-Corporate	[WPA2-EAP-CCMP]	1/1/1970 0:00	1	-88	5.6290448	-0.144389	0	5.47347	WIFI
13	64:d8:14:ee:65:62	VF-Corporate	[WPA2-EAP-CCMP]	1/1/1970 0:00	6	-79	5.6290448	-0.144389	0	5.47347	WIFI
14	64:d8:14:ee:74:d2	VF-Corporate	[WPA2-EAP-CCMP]	1/1/1970 0:00	1	-57	5.630162	-0.143993	0	22.6146	WIFI
15	0c:d9:96:22:71:33	VF-Corporate-Test	[WPA2-EAP-CCMP]	3/21/2013 12:24	1	-91	5.6299889	-0.143482	64	3	WIFI
16	0c:d9:96:39:ed:23	VF-Corporate-Test	[WPA2-EAP-CCMP]	1/1/1970 0:00	6	-73	5.5520511	-0.212232	0	13.46	WIFI
17	0c:d9:96:39:f1:b3	VF-Corporate-Test	[WPA2-EAP-CCMP]	3/21/2013 12:23	6	-86	5.6303108	-0.143852	78	3	WIFI
18	0c:d9:96:39:f2:c3	VF-Corporate-Test	[WPA2-EAP-CCMP]	3/21/2013 12:23	6	-79	5.6303215	-0.143858	78	3	WIFI
19	0c:d9:96:39:fb:43	VF-Corporate-Test	[WPA2-EAP-CCMP]	3/21/2013 12:23	6	-92	5.6303215	-0.143702	70	2	WIFI
20	0c:d9:96:4c:cd:03	VF-Corporate-Test	[WPA2-EAP-CCMP]	3/21/2013 12:22	1	-90	5.630343	-0.144131	74	2	WIFI
21	18:33:9d:3e:db:13	VF-Corporate-Test	[WPA2-EAP-CCMP]	3/21/2013 12:22	1	-92	5.6303322	-0.144158	75	3	WIFI
22	64:d8:14:b3:dd:e3	VF-Corporate-Test	[WPA2-EAP-CCMP]	1/1/1970 0:00	1	-88	5.5557597	-0.20896	0	13.46	WIFI
23	64:d8:14:ee:65:63	VF-Corporate-Test	[WPA2-EAP-CCMP]	3/21/2013 12:22	6	-88	5.6303269	-0.144276	75	3	WIFI
24	64:d8:14:ee:74:d3	VF-Corporate-Test	[WPA2-EAP-CCMP]	3/21/2013 13:08	1	-89	5.6301445	-0.143327	61	6	WIFI
25	80:fb:06:ac:4d:c8	wlanaccessv2.0	[WPA2-EAP-CCMP]	4/2/2013 13:46	6	-94	5.5655515	-0.181038	34	16	WIFI
26	00:1d:7e:29:b3:c9	2SD-FLOOR	[WPA2-PSK-CCMP]	2/26/2013 11:40	11	-91	5.5701703	-0.186596	56	4	WIFI
27	00:27:22:94:34:e5	A5A	[WPA2-PSK-CCMP]	3/21/2013 10:58	7	-94	5.5879319	-0.165315	56	2	WIFI
28	00:27:22:74:4c:5e	AIRBLAST	[WPA2-PSK-CCMP]	4/3/2013 19:59	1	-72	5.5483854	-0.201332	-14	24	WIFI
29	20:02:af:0f:d3:b6	Amlalo	[WPA2-PSK-CCMP]	1/18/2013 8:28	6	-83	5.6307131	-0.143713	76	3	WIFI
3	02:08:22:78:c3:fb	AndroidAP	[WPA2-PSK-CCMP]	3/21/2013 11:06	4	-93	5.5876851	-0.175347	76	3	WIFI
4	04:fe:31:b4:d6:da	AndroidAP	[WPA2-PSK-CCMP]	2/26/2013 13:02	6	-91	5.6258422	-0.157971	87	6	WIFI
5	28:fb:d3:85:7b:3a	AndroidAP	[WPA2-PSK-CCMP]	3/21/2013 11:31	6	-94	5.618729	-0.176082	79	6	WIFI
6	70:f9:27:c4:2c:1b	AndroidAP	[WPA2-PSK-CCMP]	2/16/2013 10:46	6	-90	5.623439	-0.176366	70	6	WIFI
7	74:45:8a:89:b5:82	AndroidAP	[WPA2-PSK-CCMP]	1/29/2013 10:13	6	-94	5.6303322	-0.092483	66	3	WIFI
8	00:13:5e:4f:68:0c	ANNEWETEY	[WPA2-PSK-CCMP]	2/26/2013 11:38	1	-88	5.5672681	-0.181215	49	3	WIFI
9	00:14:a5:b3:e3:8d	Anonymous	[WPA2-PSK-CCMP]	3/21/2013 11:27	11	-96	5.5980653	-0.179284	82	4	WIFI
10	88:32:9b:61:ca:7b	AshesGN	[WPA2-PSK-CCMP]	3/21/2013 12:22	6	-93	5.6303215	-0.14411	75	2	WIFI
11	00:1f:6d:b8:5f:72	BAGH	[WPA2-PSK-CCMP]	1/18/2013 8:21	1	-87	5.6365067	-0.129433	78	12	WIFI
12	00:1f:6d:bb:88:b2	BAGH	[WPA2-PSK-CCMP]	1/19/2013 13:25	1	-82	5.636571	-0.129277	77	2	WIFI
13	00:1f:6d:bb:8f:62	BAGH	[WPA2-PSK-CCMP]	1/29/2013 10:27	1	-91	5.6363779	-0.129535	83	8	WIFI
14	d8:c7:c8:23:ea:12	blue	[WPA2-PSK-CCMP]	3/21/2013 10:53	6	-90	5.5822831	-0.159286	46	3	WIFI
15	d8:c7:c8:23:f1:42	blue	[WPA2-PSK-CCMP]	3/21/2013 10:53	11	-96	5.5824709	-0.159377	46	3	WIFI
16	78:44:76:00:08:ca	Board Room	[WPA2-PSK-CCMP]	1/19/2013 13:26	10	-84	5.6332076	-0.137705	72	2	WIFI

Microsoft Excel - WigleWifi_20130403203507delete

	A	B	C	D	E	F	G	H	I	J	K
1	WigleWifi-1.4	appRelease=1.55	model=HTC Wildfire S A510e	release=2.3.5	device=ma	display=G	board=marv	brand=htc_europe			
2	MAC	SSID	AuthMode	FirstSeen	Channe	RSSI	CurrentLatitude	CurrentLongitude	AltitudeMeters	AccuracyMeters	Type
3	e0:5f:b9:0c:2f:a8	ciscosb	[WPA2-PSK-CCMP]	1/29/2013 10:24	6	-90	5.6413829	-0.117025	75	16	WIFI
4	4c:eb:42:54:3a:5e	Clement.idl	[WPA2-PSK-CCMP]	2/16/2013 12:00	11	-79	5.6984228	-0.290387	31	8	WIFI
5	7e:c5:37:96:ee:6e	Coldfire	[WPA2-PSK-CCMP]	2/26/2013 13:13	2	-90	5.6304449	-0.143241	76	3	WIFI
6	00:27:22:7a:5b:98	COLOUR WORLD	[WPA2-PSK-CCMP]	2/26/2013 11:37	11	-77	5.5617803	-0.171345	49	3	WIFI
7	00:23:89:d2:ad:70	COMMUNICATION	[WPA2-PSK-CCMP]	2/26/2013 11:43	1	-86	5.5709857	-0.188613	66	2	WIFI
8	00:27:22:e8:67:97	Conferenceroom	[WPA2-PSK-CCMP]	1/1/1970 0:00	9	-91	5.3755921	-0.666175	0	4578.12	WIFI
9	a0:88:b4:3b:98:a1	Connectify-BB hotspot	[WPA2-PSK-CCMP]	3/21/2013 12:23	6	-89	5.6303483	-0.143713	71	2	WIFI
10	22:df:9a:f9:d8:ad	Connectify-me	[WPA2-PSK-CCMP]	2/16/2013 9:28	1	-94	5.6397521	-0.121311	74	4	WIFI
11	44:6d:57:10:4e:9e	Connectify-me	[WPA2-PSK-CCMP]	2/26/2013 12:32	1	-93	5.5863655	-0.183823	80	6	WIFI
12	00:0c:42:e2:2d:c2	DANADAMS-EXECUTIVE	[WPA2-PSK-CCMP]	1/29/2013 10:12	1	-79	5.6282991	-0.088647	65	2	WIFI
13	00:11:24:62:74:ad	EA WiFi	[WPA2-PSK-CCMP]	1/29/2013 10:15	11	-90	5.6333792	-0.097262	70	3	WIFI
14	3c:e5:a6:12:78:a0	EPA AIR 5	[WPA2-PSK-CCMP]	4/3/2013 20:04	1	-93	5.5509818	-0.199621	57	24	WIFI
15	00:1d:7e:29:b6:ed	FINANCE ONE	[WPA2-PSK-CCMP]	2/26/2013 11:40	11	-89	5.5703688	-0.186886	57	4	WIFI
16	00:14:d1:cf:66:08	GCAAWIRELESS2	[WPA2-PSK-CCMP]	2/26/2013 12:46	5	-91	5.607217	-0.17701	103	8	WIFI
17	64:9e:f3:87:f0:18	GHP_MMC1	[WPA2-PSK-CCMP]	2/26/2013 11:40	9	-91	5.5705351	-0.187154	59	4	WIFI
18	24:db:ac:4b:e4:1d	Golden Mirathel	[WPA2-PSK-CCMP]	1/1/1970 0:00	1	-88	-?	-?	0	1630.93	WIFI
19	64:70:02:97:4f:dc	HOW-GH-OFFICE_Network	[WPA2-PSK-CCMP]	2/26/2013 9:25	4	-96	5.5885756	-0.097986	30	16	WIFI
20	76:e2:f5:23:50:58	iPad	[WPA2-PSK-CCMP]	1/19/2013 13:30	2	-86	5.6313944	-0.141985	67	2	WIFI
21	00:13:5e:51:88:25	JubailiBros2	[WPA2-PSK-CCMP]	1/19/2013 13:27	1	-94	5.6329501	-0.138359	72	3	WIFI
22	50:cc:f8:95:ca:5f	KobasDroidSpot	[WPA2-PSK-CCMP]	3/21/2013 11:07	11	-95	5.588581	-0.179687	76	4	WIFI
23	00:13:5e:4f:90:84	KOFORINET	[WPA2-PSK-CCMP]	1/19/2013 13:17	6	-85	5.6340927	-0.098485	68	4	WIFI
24	3c:e5:a6:22:01:61	M.F.A 2	[WPA2-PSK-CCMP]	2/26/2013 12:28	1	-90	5.579955	-0.186306	73	3	WIFI
25	00:27:22:94:7d:e5	MBSHQ-2	[WPA2-PSK-CCMP]	2/26/2013 11:32	6	-89	5.5622792	-0.145376	41	3	WIFI
26	5c:0a:5b:44:77:81	megzycan	[WPA2-PSK-CCMP]	1/29/2013 10:24	6	-91	5.6407338	-0.118956	74	12	WIFI
27	3c:e5:a6:12:75:80	Min of Employment 02	[WPA2-PSK-CCMP]	4/3/2013 20:05	1	-91	5.5512124	-0.195962	60	6	WIFI
28	bc:76:70:66:cd:c0	MINIPOLICE	[WPA2-PSK-CCMP]	4/3/2013 20:04	4	-89	5.55103	-0.198913	54	12	WIFI
29	00:23:89:de:60:80	MOC1	[WPA2-PSK-CCMP]	4/3/2013 20:05	1	-97	5.551073	-0.197947	54	6	WIFI
30	3c:e5:a6:12:78:20	MOC1	[WPA2-PSK-CCMP]	4/3/2013 20:05	1	-92	5.5511051	-0.197223	53	8	WIFI
31	00:23:89:de:60:81	MOC2	[WPA2-PSK-CCMP]	4/3/2013 20:05	1	-95	5.551073	-0.197947	54	6	WIFI
32	3c:e5:a6:12:78:21	MOC2	[WPA2-PSK-CCMP]	4/3/2013 20:05	1	-90	5.5511051	-0.197223	53	8	WIFI