International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 01 | Jan 2019 www.irjet.net p-ISSN: 2395-0072

Analysis of Security Vulnerabilities in Wifi-Protected Access Pre-Shared Key (WPA-PSK/ WPA2-PSK)

Michael Asante[1], Kwabena Akomea-Agyin[2]

[1,2]*Department of Computer Science, Kwame Nkrumah University of Science and Technology, Kumasi, State, Ghana*

Abstract:- *This research paper analyzed the security vulnerabilities within the architecture of WPA/WPA2-PSK that can be used to hack into a WPA/WPA2-PSK enabled wireless network.*

It was found out that WPA-PSK and WPA2-PSK use a four-way Extensible Authentication Protocol Over LAN (EAPOL) handshake to exchange information leading to the installation of private keys for data encryption. These private keys are called the Pairwise Master Key (PMK) and the Pairwise Transient Key (PTK).

It was also found out that the four-way handshake are plaintext. Hence a hacker could easily eavesdrop on them and retrieve the ANounce, SNounce, Authenticator Mac Address, Supplicant Mac Address, and the MIC messages.

The only thing that is unknown is the Passphrase used by the legitimate user of the network. It was proving that if an attacker can guess the passphrase or find it within a dictionary, the attacker can compute the PTK for each guess and use the MIC to verify that the guess is correct. The attacker will end up deriving the Passphrase to the legitimate network if and only if the passphrase exists in the attacker's dictionary.

At the end of the study, it was concluded that WPA-PSK or WPA2-PSK is safe as long as the Passphrase to the network cannot be found in the attacker's dictionary of possible passphrases.

A survey was conducted involving 1,271 Access Points (APs) in Ghana. It showed that 8.3% of the surveyed networks were using WPA-PSK with TKIP. 4.3% were using WAP-PSK with CCMP. 7.9% were using WPA2-PSK with TKIP and 49.2% were using WPA2-PSK with CCMP. All these surveyed networks are vulnerable to WPA-PSK / WPA2-PSK dictionary attacks if the passwords are simple and can be found in the attacker's dictionary.

Keywords: Wifi-Protected Access (WPA), Pre-shared key (PSK), Extensible Authentication Protocol Over LAN (EAPOL), Pairwise Master Key (PMK), Pairwise Transient Key (PTK), Temporal Key Integrity Protocol (TKIP), Counter Mode with CBC MAC Protocol (CCMP).

1. Introduction

Recall from the paper on WEP (ANALYSIS OF THE SECURITY VULNERABILITIES IN WEP) that WEP suffered from the following vulnerabilities:

1) The use of the static keys for both authentication and encryption. Thus once the key is compromised during authentication, the same key can be used to decrypt every packet.
2) There is no mutual authentication. Only the AP authenticates the client. Thus the client has no way of proving the legitimacy of the AP.
3) The use of short IV space (24 bits) which leads to IV reuse and keystream reuse attacks.
4) The use of a linear checksum ICV which leads to message injection and modification attacks. Messages can be modified and still ensured that the resulting checksum is still a valid checksum without detection.

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 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 based on the Advanced Encryption Standard (AES) block cipher (Halvorsen & Haugen, 2009).

1.1 Architecture of TKIP

WPA-TKIP defined 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.

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.

1.1.1 TKIP Packet Structure

TKIP makes some modifications to the WEP packet structure (Halvorsen & Haugen, 2009). The TKIP packet structure is shown in figure 1.

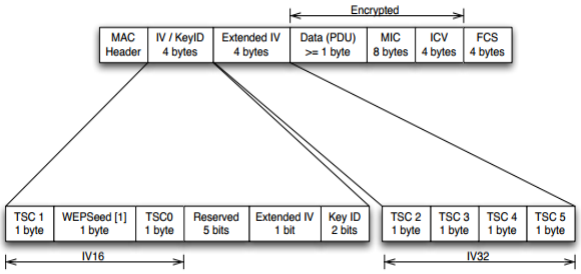

Figure 1: TKIP Packet structure

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.

1.1.2 TKIP Sequence Counter (TSC)

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

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

A 48-bit TSC address 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.

1.1.3 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).

1. The secret authentication key k is the Pairwise Temporal Key (PTK) which is generated from the Pairwise Master Key (PMK).
2. 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.
3. The sender sends the message M and the generated MIC to the receiver.
4. The receiver computes the PTK (k) of the received message M using its PMK, and generates an MIC for the message.
5. The receiver's computed MIC and the sender's MIC acts as input into the verification predicate.
6. 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 1 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.

1.2 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, none of the attacks described in WEP or TKIP will work against it. As at the time of writing this thesis, there are no known practical attacks against CCMP or AES, except from brute-force attacks on the Extensible Authentication Protocol over LAN (EAPOL) handshake.

Figure 2 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.

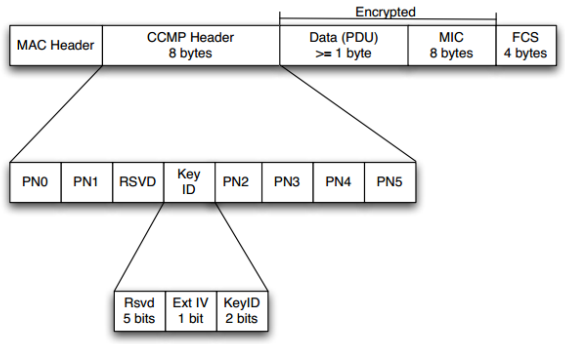

Figure 2: CCMP Packet Structure

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:

Table 1: 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*

1.3 WPA/ WPA2-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.

1.3.1 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 Asymmetric or Public Key Cryptography as discovered by Diffie and Hellman in their book "New Directions in Cryptography" (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 passhrase 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 3.

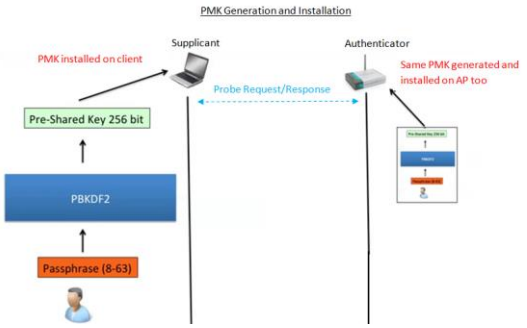

Figure 3: PMK Generation and Installation mechanism

1.3.2 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.

1.3.3 Pairwise Transient Key (PTK)

In order to obtain the PTKs or GTKs, a four-way and 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).

1.3.4 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 4. The ANounce is sent as part of Message 1 to the client in figure 5.

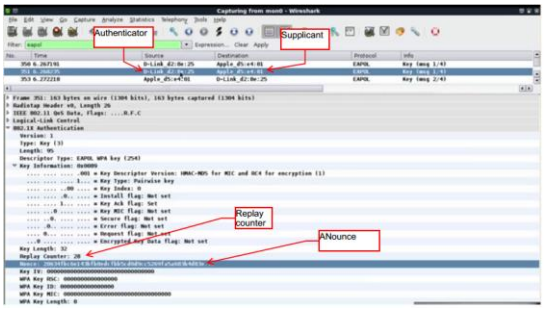

Figure 4: A random ANounce sent by the authenticator to the supplicant

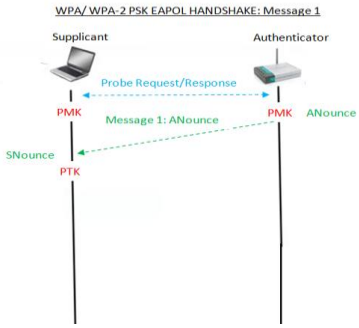

Figure 5: Message 1 of the EAPOL Handshake sent from the Authenticator to the Supplicant

1. 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 5.
2. 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).
3. 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 6.

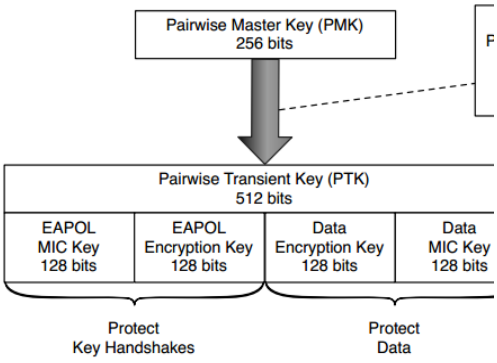

Figure 6: The functional parts of the 512 bits generated PTK

4. 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 7 and 8.

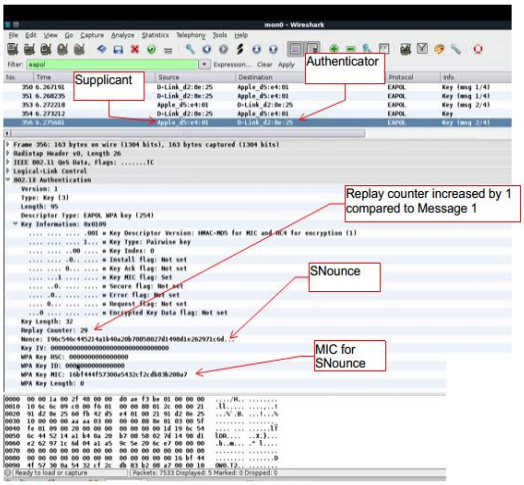

Figure 7: A random SNounce sent by the supplicant to the authenticator

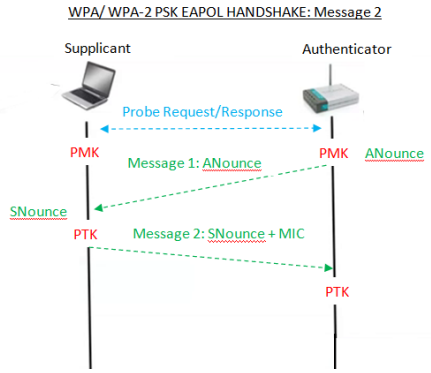

Figure 8: Message 2 of the EAPOL Handshake sent from the Supplicant to the Authenticator

5. 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).

6. 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.

7. 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.

8. Next the authenticator sends Message 3 which is the Key installation message as shown in figures 9 and 10 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.

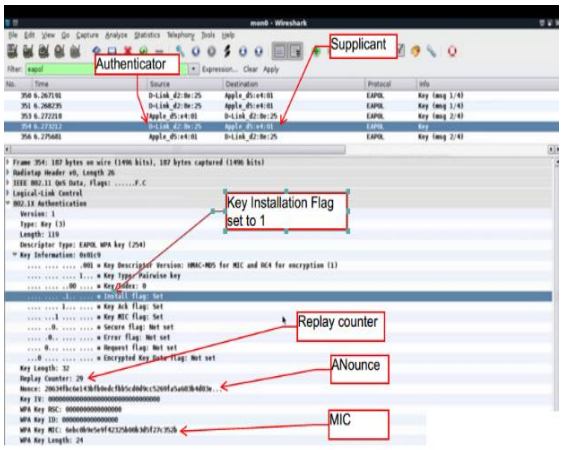

Figure 9: Message 3 of the EAPOL Handshake sent from the Authenticator to the Supplicant as captured with Wireshark

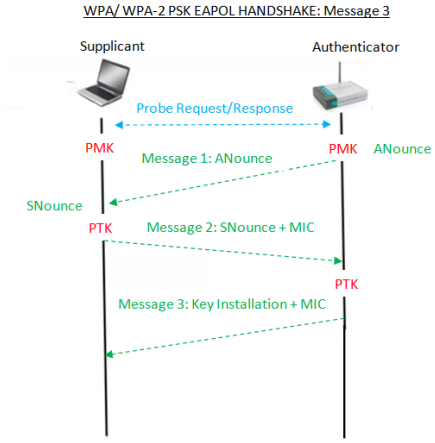

Figure 10: Message 3 of the EAPOL Handshake sent from the Authenticator to the Supplicant

9. 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 11.

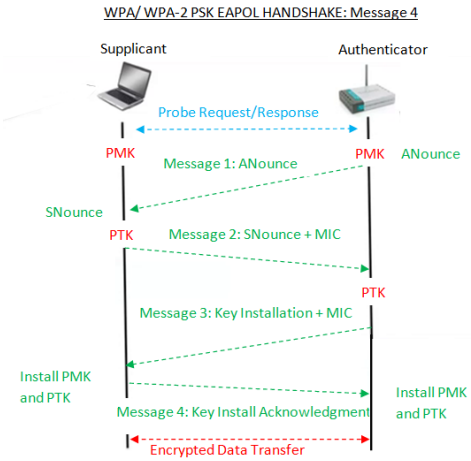

Figure 11: Message 4 of the EAPOL Handshake sent from the Supplicant to the Authenticator

10. 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 11.

2. Methodology

A laboratory was setup that included a victim machine (supplicant), an Access Point, an Authentication server, a hacker machine (running BackTrack 5) and an Alfa AWUS036NH wireless card connected to the attacker's machine as shown in figure 12.

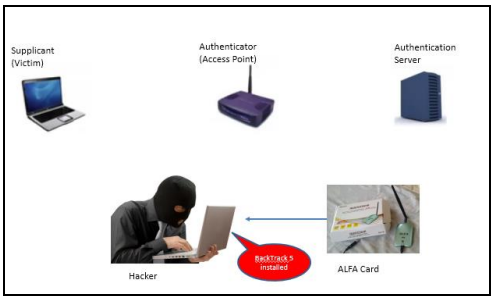

Figure 12:The laboratory setup diagram

3. Vulnerabilities in IEEE 802.11 WPA/WPA2-PSK Security Protocol

Upon studying and analysis literature on the architecture of the IEEE 802.11 WPA/ WPA-2 PSK security protocol, the following three vulnerabilities were discovered:

3.1 The four-way EAPOL Handshake are not encrypted over the air interface:

The four-way EAPOL handshake 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 4 to 7.

3.2 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)*.

3.3 There is an MIC (Plaintext) to ensure that the computed PTK is the same as the one computed by the legitimate client:

The 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

4. Analysis of the vulnerabilities in IEEE 802.11 WPA/WPA2-PSK Security Protocol

4.1 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.

4.2 Message 2 of the EAPOL handshake which contains the SNounce and 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.

4.3 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.

4.4 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.

4.5 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.

Significance:

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.

These vulnerabilities led to successfully cracking the WPA/ WPA-2 PSK passphrase.

5. 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 13: The WPA/ WPA-2 PSK passphrase was "Ashes@112".

Figure 13: 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 14. The client then connected to the access point.

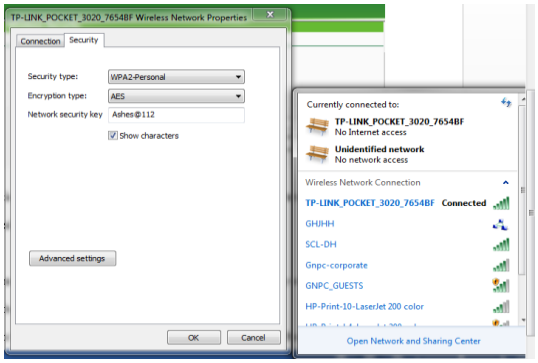

Figure 14: The legitimate client configured to support WPA/ WPA-2 PSK with same password as the AP

3. "airodump-ng mon0" command was used to monitor the WPA/ WPA2-PSK network called TP-LINK_POCKET_3020_7654BF as shown in figure 15. It cipher suite was CCMP.

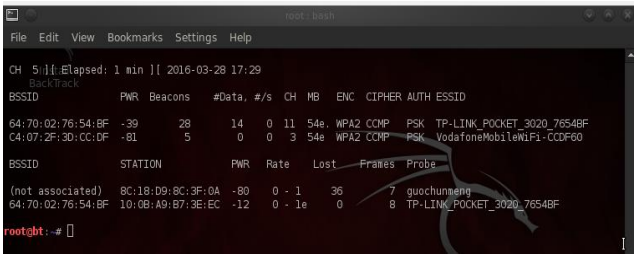

Figure 15: The output of "airodump-ng" used to monitor broadcasting SSIDs

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 16:

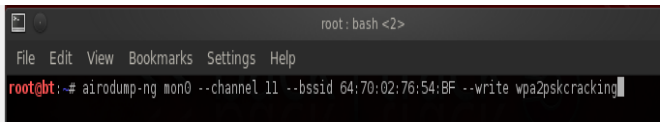

Figure 16: The eavesdropping and saving of the PSK EAPOL Handshake

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 17.

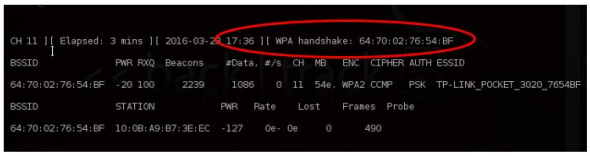

Figure 17: The capturing of the four-way EAPOL handshake as soon as a legitimate client connects to the legitimate AP

6. Next a default dictionary file directory in Backtrack5 was opened and edited to include additional potential passwords as shown in figures 18 and 19.

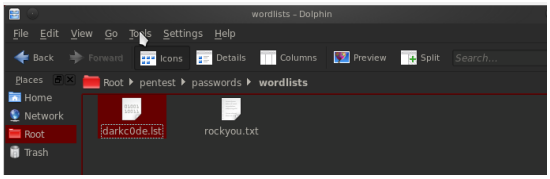

Figure 18: The location of a default dictionary file in BackTrack5

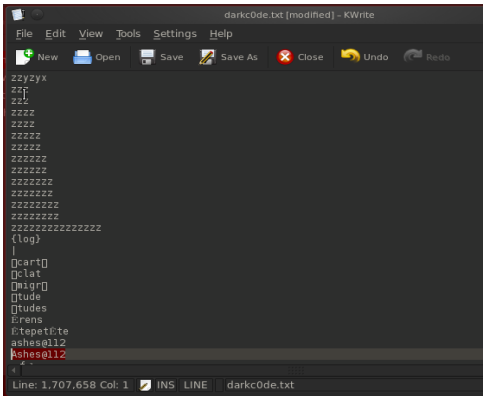

Figure 19: The contents of the default dictionary file edited to include "Ashes@112" passwords

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 20:

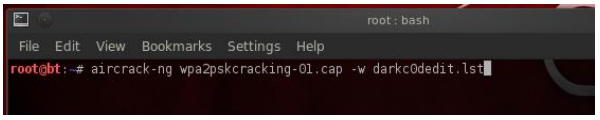

Figure 20: "aircrack-ng" fed with the captured four-way handshake file and a dictionary file

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 21. 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.

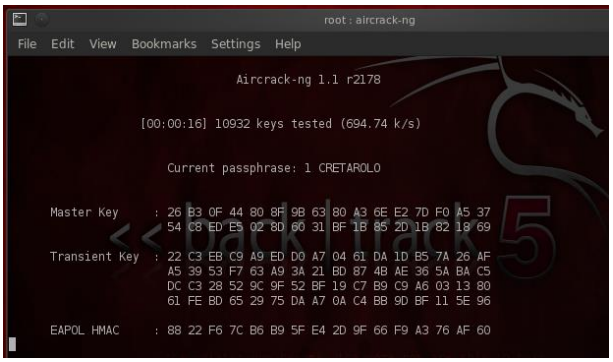

Figure 21: "Aircrack-ng" trying various combinations of passphrase in an attempt to crack the key

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 22.

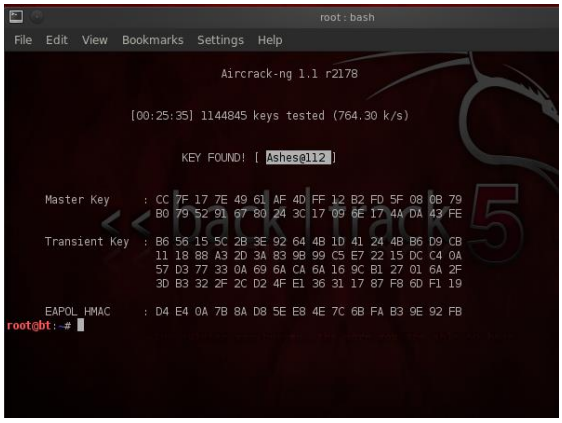

Figure 22: "Aircrack-ng" locating the correct passphrase and hence cracking the WPA/ WPA-2 PSK key

Analysis:

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

6. Research Findings

1. WPA/ WPA2-PSK Passwords can only be cracked if the password can be found in an attacker's dictionary.
2. The attacker's dictionary is editable.
3. The attacker's dictionary is case-sensitive.

7. Conclusion

From this thesis work, it have proven that there are indeed vulnerabilities in WPA/ WPA2-PSK security protocol. These vulnerabilities can be easily exploited by an attacker to gain unauthorized access into a Wireless Local Area Networks that has been secured with WPA/ WPA2-PSK.

8. Recommendation

1) 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.
2) The attacker's dictionary is editable. It is recommended that users do not use passwords that can be found on the internet.
3) 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.

9. Future work

Future work includes conducting further study into WPA/WPA-2 EAP to identify if there are any vulnerabilities that can be used to compromise a WPA/ WPA-2 EAP enabled network, finding vulnerabilities in WLANs as they are prone to attacks using Man-in-the Middle Attacks, Denial of Service Attacks, patching the flaws in the WPA/ WPA-2 security protocols of WLANS, investigating and the development of a robust and secured centralized management solution for large enterprises and also investigation into the pros and cons of Network Intrusion Detection Systems(NIDS).

References

[1] IEEE, IEEE 802.11 Standards documents. http://standards.ieee.ogr/wireless/. January 2004.

[2] Rackley Steve, "Wireless Networking Technology. From Principles to Successful Implementation". May 2007.

[3] Vivek Ramachandran. "BackTrack 5 Wireless Penetration Testing. Biginners Guide". ISBN 978-1-849515-58-0. September 2011.

[4] 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.

[5] Winget C.N, Moore T, Stanley D, & Walker J. "IEEE 802.11i Overview". September 2007.

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

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

[8] Akin Devin. "802.11i Authentication and Key Management (AKM)". Whitepaper. Certified Wireless Network Professional. Planet3 Wireless Inc. May 2005.

[9] Walker Jesse. "802.11 Security Series. Part II. The Temporal Key Integrity Protocol (TKIP)". Network Security Architect. Platform Networking Group. Intel Corperation. June 2005.

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

[11] 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.

BIOGRAPHIES

Michael Asante is a Senior Lecturer in Computer Science and holds a PhD in Systems Engineering from the University of Reading and also Master of Science in Scientific Computing and Information Technology from the London South Bank University, London all in the United Kingdom. His areas of

Specialization are Data Communication, Computer Networks and Security and Distributed systems.

Michael's research interests are in the areas of Network Protocols, Network Architectures, Wireless Networks, Mobile Internet Protocol Convergence and Ubiquitous Systems, Network Security, Distributed Systems, Computer System Security, Multimedia Compression Techniques and Standards, Wireless Multimedia Transmission, Cross- layer Optimization and Voice-over Internet Protocol.

Kwabena Akomea-Agyin is a cofounder NSA Consortium, a company that specializes in preventing and responding to network security incidents. Kwabena is a strong evangelist of security with about 8 years working experience in managing networking, security, and transmission portfolios within the telecommunication space. His research interest is in digital forensics and cyber security as a defensive tool to helping organizations and individuals achieve confidentiality, integrity, and availability over the intranet and internet. Kwabena holds an MSc in Information Technology from Kwame Nkrumah University of Science and Technology.