Practical WLAN Security

Jakob Petersson Gustaf Ouvrier
Email: { jakpe547, gusou347 }@student.liu.se
Supervisor: Ulf Kargén, {ulf.kargen@liu.se}
Project Report for Information Security Course
Linköpings universitet, Sweden

Abstract

This report concerns the security available for the protection of Wireless Local Area Networks. First some background knowledge of the IEEE 802.11 network standard and architecture are presented. Then the current security standards related to wireless networks (WEP, WPA, WPA2) are described and reviewed with focus on vulnerabilities allowing for key recovery attacks. Furthermore Wi-Fi Protected Setup (WPS) is explained and its security issues described.

In addition, demonstrations of two practical attacks are presented. The first attack is performed against a WPA2 protected network and seeks to recover the key using a dictionary. The second attack shows how to exploit the flaws in WPS to brute force the PIN code and thereby obtain the key.

1. Introduction

Today wireless networks are common in almost every home and wireless communication capabilities are an essential part of all new mobile devices. It is even becoming available in everyday consumer electronics which uses it to connect to the internet to enhance its functionality. The main reason for its wide use is the convenience of the wireless communication itself; being able to connect just about anywhere and anytime without the hassle of cables. But by communicating through the air, which is a shared medium, your communications become easily accessible to everyone in your vicinity. This introduces an array of security concerns that does not exist in wired communication. For Wireless Local Area Networks, which we will focus on in this article, there exist cheap and easily acquired equipment that an attacker can use to tap into any wireless channel threatening both the confidentiality and integrity of a communication.

This article will begin by introducing some background knowledge needed to understand the concepts used in the rest of the report. It will then proceed by describing the progress of WLAN security protocols, starting with WEP, WPA and then WPA2. It describes how they work and how each new protocol differ and improve the security in relation to the previous. For each protocol known weaknesses and attacks will also be described with focus on attacks that recovers the key of the network. Furthermore, the article will explain what WPS is and how its major design flaws made it possible to bypass the security provided by WPA and WPA2. Finally the article will demonstrate two practical examples of a key recovery attack against a WPA/WPA2 protected network and WPS respectively.

2. Basic WLAN Security

This section will introduce and explain some concepts that are important for understanding the rest of the report.

2.1 WLAN Standard IEEE 802.11

Wireless Local Area Networks (WLANs) is a technique that allows two or more devices to communicate with each other over the air and it is normally used to provide access to a larger wired network such as a company network or the internet. The specifications for implementing a WLAN is specified and maintained by the *Institute of Electrical and Electronics Engineers* (IEEE) and are defined in the standard 802.11: Wireless LAN Medium Access Control and Physical Layer Specifications [1].

Since IEEE 802.11 was introduced in 1997 many variations have been added in form of amendments to the original standard. These additions include adding more functionality, improving the transfer speeds, allowing for the use of more frequencies and more. The standard also include protocols that provides security and these are what we will focus on in this paper.

2.2 Architecture

All devices that can connect to wireless network using a wireless network interface are called stations. A station can be either a client or an *Access Point* (AP). A client refers to all wireless enabled devices such as laptops, smartphones, tablets, etc. while APs are base stations for the wireless network. All stations has a unique identifier

called a *Media Access Control* (MAC) address that is used to identify them and specify the receivers and senders of packets sent between them.

A set of stations connected to each other is referred to as a *Basic Service Set* (BSS) and a set of BSSs is called an *Extended Service Set* (ESS). A BSS is identified by the APs MAC address and is called an BSSID. In an ESS all APs are connected by a distribution system and is identified by a unified ID called the *Service Set Identification* (SSID). The SSID is also referred to as the name of a WLAN and is used by a client when it wants to join the network.

A network using the IEEE 802.11 standard can operate in two basic modes: ad hoc mode and infrastructure mode. When operating In ad mode the mobile units communicate directly with each other, while in infrastructure mode they communicate through an AP.

3. WEP – Wireless Equivalent Privacy

Wireless Equivalent Privacy (WEP) was introduced with the original IEEE 802.11 standard ratified in 1999 and as the name suggests its purpose was to provide security equivalent to that of a wired network. In order to achieve this, WEP was designed with three main security goals in mind: confidentiality, access control and integrity. To provide the required confidentiality the communication is encrypted to prevent eavesdropping. To protect the wireless network from unauthorized access the standard includes an optional feature to only accept packets that are properly encrypted, and to preserve the data integrity all message have a integrity checksum.

The encryption standard used in WEP is a stream cipher called RC4. A stream cipher is a symmetric key cipher where a message is combined with a generated pseudorandom cipher digit stream, referred to as the keystream, one digit at a time. In order to generate the keystream a seed value is used as a cryptographic key to define the initial state of the cipher. The following states are then serially computed based in the previous states, resulting in a pseudorandom keystream that can be computed by both the sender and receiver. This is called a *Pseudo Random Number Generator* (PRNG).

Due to US government export restrictions WEP was first designed to use a 64 bit key. This key length was at the time considered short enough to allow practical brute force attacks with fairly modest computing resources. When the restrictions were lifted, manufactures implemented WEP with a much longer 128 bit key.

In WEP the key is divided into two parts: a static and a dynamic part. The static part of the key is the secret cryptographic key that is shared between everyone connected to the network. The dynamic part of the key is a randomly selected 24 bit value called the *Initialization Vector* (IV). The IV changes after every message and its purpose is to ensure that each message is encrypted using a different key.

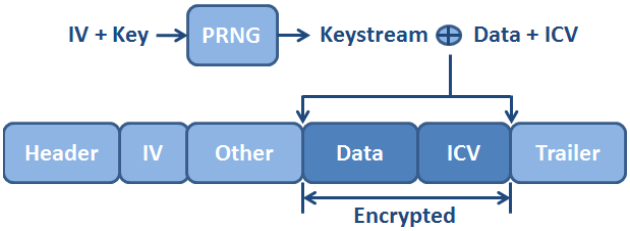

Figure 1: WEP encryption process

When a packet is sent the message plaintext is first combined with its integrity checksum called *Integrity Check Value* (ICV) calculated by an error-detecting code called CRC-32. This is then encrypted using the RC4 algorithm and the output of the encryption is sent to the receiver along with the IV. The IV is sent in clear text so that the receiver can combine it with its copy of the shared cryptographic key and reverse the encryption to reveal the original plaintext message. [1] [2] [3]

3.1 WEP Weaknesses

Even though WEP sought to provide the same level of security as a wired network it eventually became apparent that there were serious design flaws in the protocol that made attacks on WEP possible.

3.1.1 Key Recovery

In 2001 Fluhrer, Mantin and Shamir published an article [4] in which they described an attack that can recover the RC4 key by passively eavesdropping on the network. This attack was made possible due to a weakness caused by the way that the RC4 cipher and IVs are used in WEP. In WEP the IV is sent in the clear and the IV forms the beginning of the key used in RC4. What Fluhrer et al. discovered is that the first few bytes in the keystream are strongly nonrandom resulting in certain IV values producing weak WEP keys. When a weak key is used to encrypt a message the first part of the keystream may contain some correlation with the secret key, which has a small probability of leaking information about the key. By collecting enough packets the whole secret key can be recovered.

A couple of years later in 2005, Andreas Klein presented another article [5] on weaknesses in the RC4 stream cipher when used in WEP. In the article he described the existence of even more correlations between the keystream and the key than what Fluhrer et al. had discovered previously. These discoveries then became the base for a much faster attack in 2007, called the PTW attack [6] after its creators Pyshkin, Tews and Weinmann. Using this attack they were able to recover the secret key with an order of magnitude less captured packets. It was also possible to accelerate the

attack by performing deauthentication and packet injection. Using packet injection means that you resend captured packets or send your own fabricated packets into a network without being connected to it. In a deauthentication attack this is utilized to fabricate disassociation frames and send them to a connected client, forcing it to reauthenticate. Both of these techniques were utilized to greatly increase the number of interesting packets sent over the network, making it possible to complete the attack in less than 60 seconds.

3.1.2 Other Weaknesses

Apart from actual key recovery, WEP also has other weaknesses. Due to the fact that the only part of the key that changes is the IV it does not take long until the same key is reused. If the keystream for a given IV is found, an attacker can decrypt subsequent packets that were encrypted with the same IV or forge packets. Furthermore, the checksum WEP uses to ensure the data integrity is in itself weak. CRC-32 is based on a linear function which makes it possible to make changes in the message and then correctly adjust the checksum making the message look valid again. [2] [3] [7]

In the wake of these discoveries, several tools exploiting the flaws were developed and made publicly available and as a result WEP eventually became deprecated in favor of WPA and WPA2.

4. WPA(2) – Wi-Fi Protected Access

This section will describe how WLAN security has progressed after WEP, describing improvements in the newer standards WPA & WPA2 as well as their weaknesses and known attacks against them.

4.1 WPA

After it became clear that WEP had serious security issues the Wi-Fi Alliance together with IEEE started development of the more secure WLAN standard IEEE 802.11i. But the industry could not wait for this standard to be finalized so an interim solution called *Wi-Fi Protected Access* (WPA) that implemented a subset of the 802.11i standard and targeted all known WEP vulnerabilities was developed and released in early 2003. Since WPA was meant as a fix to the problems in WEP it was designed so that only a software or firmware upgrade was necessary to secure existing and legacy hardware and this meant that some of the enhancements of WPA had to build upon existing WEP functionality. WPA introduced the *Temporal Key Integrity Protocol* (TKIP) to generate per packet keys and improved the message integrity checksum with a new *Message Integrity Protocol* (MIC) called Michael. It also added a 4-way handshake, improved the IV, introduced the *pairwise key hierarchy* and two different ways of authentication. [3] [7]

Both WPA and WPA2 allows for two modes of operation, Personal mode and Enterprise mode. In personal mode a *Pre-Shared Key* (PSK) in the form of a passphrase is used to provide mutual authentication between a connecting client and the AP. In Enterprise mode authentication is instead done by a RADIUS authentication server using the 802.1X protocol. This allows for centralized authentication in the network and in this mode no PSK is used but instead a new key called the *Master Session Key* (MSK) is generated for the client after successful authentication. [11]

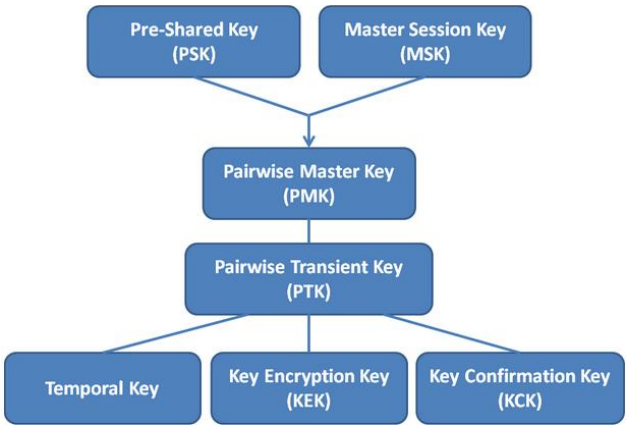

Figure 2: Key hierarchy

Depending on the type of operation either the PSK or the MSK is used by the PBKDF2 [23] algorithm to create another key called the P*airwise Master Key* (PMK). The PMK serves as a main key from which temporal keys such as session key, group key etc. is derived reducing the exposure of the secret which was a big flaw in WEP. [1] [11]

The 4-way handshake that WPA introduced occurs after a client is associated to an AP and the PMK has been created. During this handshake the PMK is used to derive a session key called *Pairwise Transient Key* (PTK) which is a container for three keys: The *Temporal Key* (TK) which is used to encrypt normal communication on network, and the *Key Confirmation Key* (KCK) and *Key Encryption Key* (KEK) which are used for encryption and integrity when delivering temporal keys to and from the AP [11]. See detailed description of the handshake in section 4.3.1.

The TKIP uses the session key as a seed value to generate per-packet keys. Each packet's key is created using a mixing function that generates the key by hashing the sender's MAC address, the IV, and the session key. This results in an effective 128-bit dynamic key. Compared to WEP, TKIP also improves the length of the IV from 24 to 48 bits and forbids the use of known weak IVs. To prevent replay attacks TKIP also uses the IV for packet sequencing and to ensure message integrity and protect against forgery attacks it applies the message integrity code Michael.

Michael is a hashing function that calculates a 64-bit value across the entire raw data packet before it is encrypted. If a WLAN detects a modified message it will trigger countermeasures in the form of forcing each device on the network to request of a new session key as well as disable the wireless link for 60 seconds. To minimize the exposure all temporary keys are also changed after every 10 000 packet. [3] [7]

4.2 WPA2

The full 802.11i standard was finished and released in 2004 and it is implemented in the security protocol WPA2, also referred to as *Robust Secure Network* (RSN). WPA2 is the latest and currently most secure WLAN security protocol. Unlike WPA, WPA2 is not built upon the WEP architecture and thereby requires new hardware to work. WPA2 adds support for a new and stronger encryption standard named *Counter Cipher Mode with block chaining message authentication code Protocol* (CCMP) meant to replace TKIP which still relies on RC4. CCMP is instead based on the *Advanced Encryption Standard* (AES) and uses a 128 bit key and 128 bit block size. [3] [7] [11]

4.3 WPA and WPA2 Weaknesses

WPA and WPA2 did an excellent job of patching the problems that existed in WEP. The improvements practically nullified the exposure of the PSK during communication and when using 802.1X authentication the key is practically impossible to recover. However, a security weakness allowing for the recovery of the PSK from a network operating in personal mode still exists. The weakness lies in how the AP and the connecting client calculates the PTK during the 4-way handshake.

4.3.1 4-Way Handshake

The AP starts by generating an *Authentication Nonce* (ANonce) and sends it in the clear to the client. The client in turn generates a *Supplicant Nonce* (SNonce) and derives the PTK using the nonces. The client then sends the SNonce in the clear and a MIC encrypted with the KCK back to the AP. As mentioned earlier the KCK and KEK are parts of the PTK. The AP derives the PTK itself and sends a *Group Temporal Key* (GTK) and a MIC also encrypted with KCK to the client. The client verifies the PTK by checking that the MIC was encrypted with the same KCK and sends an acknowledgement encrypted with KEK together with a MIC encrypted with KCK back to the AP. The AP finally verifies the PTK and the 4-way handshake is complete. Note that both nonces are sent in the clear. [1] [9]

4.3.2 Key Recovery

The goal of the attack is to recover the PSK and from earlier we know that the PSK is used to derive the PMK using a known algorithm called PBKDF2. The algorithm takes the PSK, SSID and length of SSID as input and involves hashing 4096 times. The PMK is then used together with ANonce, SNonce, client MAC address and AP MAC address to derive the PTK using a PRNG. Some vendors provide the feature of hiding the SSID, but this does not hinder it from being revealed when the client connects and as noted previously the nonces needed to derive the PTK is sent in the clear. This leaves only the PSK unknown and by capturing the 4-way handshake an attacker gets all the information needed to calculate the PTK, trying different PSKs. [8] [10] [11]

In practice a brute force or dictionary attack against a WPA or WPA2 protected network consists of three steps. First the 4-way handshake is captured by either passively monitoring the network traffic or by using a deauthentication attack to force a connected client to reauthenticate. Then a PSK is guessed and its corresponding PTK is computed according to the captured handshake. Finally the KCK encrypted MIC value is calculated and checked against the MIC in the captured handshake. If they are equal the guessed PSK is correct and the attack is successful, otherwise a new guess is processed. The rate at which guesses can be processed using brute force or a dictionary is rather low due to all the calculations required by the PBKDF2 algorithm. To speed up the guessing process during an attack, precalculation of the PMK can be done for common SSID's and PSKs and be stored in so called rainbow tables. The result can then later be used to attack networks using those SSIDs and if the PSK matches any of the ones stored the key recovery process is much faster [12]. So in essence the strength of the protection provided by WPA and WPA2 lies in the strength of the PSK. The strongest protection is gained by having a long and random PSK and an unusual SSID, and the weakest by having both a common SSID and a popular PSK that exist in dictionaries or rainbow tables. A practical example of this kind of attack is described in section 6.2. [8] [9] [10]

4.3.3 Other Weaknesses

Apart from actual key recovery, there are other weaknesses in WPA and WPA2. Even though WPA2 includes the stronger encryption option CCMP, some WPA2 and all WPA protected networks still uses TKIP and are thereby exposed to its weaknesses. In 2008 Tews and Beck published an article [13] in which they describes a flaw in TKIP that allows an attacker to inject up to 7 custom packets into a network after successfully decrypting an ARP message. This attack was later optimized and improved upon by Yosuke et al. in 2012 [15], and Vanhoef et al. in 2013 [16].

Networks protected by WPA2 are vulnerable to the Hole196 attack [14]. In the attack a malicious authorized user, aka insider, exploits the GTK to sniff and decrypt data of other users on the network and may even install malware and compromise their devices.

5. WPS – Wi-Fi Protected Setup

This section will describe what WPS is, why it was introduced, what weaknesses came with it and how it can be used to bypass the security provided by WPA and WPA2.

5.1 WPS

Wi-Fi Protected Setup (WPS) was introduced by the Wi-Fi alliance in 2007 and its purpose is to make it easier for normal home and business users to configure the security on their WLAN and connect new devices to an existing network without the user having to type in a long passphrase. There are two main methods in WPS by which this can be done: Using a pushbutton or using an 8-digit PIN. When using a pushbutton the user is required to have physical access to the AP and push a real (or virtual) button on both the connecting client device and the AP within a certain amount of time. The PIN option can be used in two different ways, either the PIN of the client device is entered into the APs web interface or the PIN of the AP is entered into the client device. After performing one of these tasks successfully the AP provides the client device with the PSK making it able to connect to the network.

5.2 WPS Weaknesses

Even though WPS was designed to provide a secure way of configuring a WLAN, the protocol has serious design flaws that makes it vulnerable to key recovery attacks.

As described previously a client can connect to an AP using no additional authentication than providing the PIN of the AP. Unlike the two other methods which use the pushbutton or web interface this method does not require physical access or an established connection to the AP which makes it possible to attempt a brute force of the PIN from outside the network.

The last digit in the PIN is a checksum calculated from the other digits which means that the actual PIN is only 7 digits long. However, the length of the PIN is still too large to complete a brute force attack within a reasonable amount of time. To make a brute force attack viable the number of PIN attempts has to be reduced. In 2011 Stefan Viehböck [17] and Tactical Network Solutions [18] independently discovered a flaw in the WPS protocol that reduced the number of brute force attempts needed from 10^7 to only 11 000.

During the WPS-authentication eight messages are sent between the client and the AP. The first two messages are used to establish a shared encryption key for the communication using Diffie-Hellman key exchange. Messages three through seven are then used to verify that both parties have the correct PIN. This is done in two steps, verifying half of the PIN in each step. Finally messages seven and eight are used to send configuration data. If the authentication fails at any point, the AP will send a NACK message and end the authentication process.

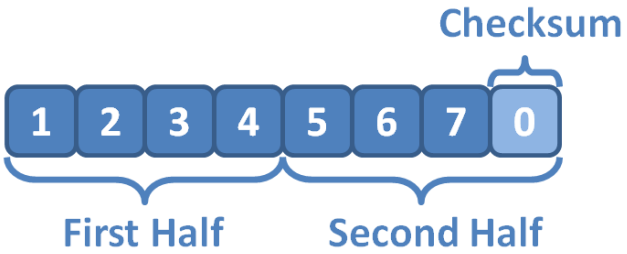

Figure 3: WPS PIN

The flaw in the protocol is that the PIN is divided into two halves which are sent separately in different messages and the fact that the authentication process is ended as soon as one message is wrong. This means that the AP will send a NACK directly if the first half of the PIN is incorrect and thereby allowing the attacker to test the first half the PIN separate from the second. As a result only $10^4 + 10^3 =$ 11 000 brute force attempts are needed, making it a feasible attack. A practical example of this kind of attack is described in section 6.3.

Even though these design flaws made it practical to mount a brute force attack, measures can be taken to mitigate the risk of these flaws being exploited. For the attack to be practical the attacker needs to be able to test a large number of PINs in a reasonable amount of time, which means that a simple lockdown period after a few failed attempts can slow down the attacker. Another more preventive approach is to instead fully disable the WPS functionality after a number of failed attempts.

6. Practical Attacks against WLAN

This section describes the two practical attacks we performed against WPA and WPA2 as well as WPS.

6.1 Preparations

To perform a practical attack against a real network we needed both hardware and software that was suitable for the task. A laptop running *Kali Linux* [19] inside a virtual machine was used as the platform to launch the attacks from and a target network was hosted by a normal home router from D-Link of model DIR-615. This router was chosen because it supported all the protocols we wanted to demonstrate an attack upon.

Since *Kali Linux* is a Linux distribution intended for use in penetration testing and therefore comes equipped with many of the drivers and tools needed to perform attacks on WLAN we decided it was very well suited for our purposes and needs. All of the tools we used came installed in the live DVD of *Kali Linux* that is available from their website. This included the *aircrack-ng suite* [20], a series of tools used for wireless network auditing and *reaver* [21], a tool used to brute force a WPS pin using the weaknesses described earlier.

Kali Linux is a capable collection of software but we also needed a suitable wireless card to be able to interact with the target network. The wireless card we chose to use for the attacks is a simple USB dongle that utilizes the *Realtek rtl8187* driver and it is supported by *Kali Linux* right out of the box. The reason why chipsets and drivers are of importance is because they have different features and varying levels of support from the individual tools. In order to perform the attacks we needed a wireless card that supported both monitor mode and packet injection. Monitor mode is a mode in which a wireless card can passively listen on wireless communication and capture packets without being connected to a network. Similarly, to support packet injection means that a wireless card is capable of sending (or injecting) its own packets into a network without being connected to it.

6.2 Attack on WPA(2) PSK

The attack on WPA and WPA2 demonstrated in this section is based on the attack that was discussed earlier in section 4.3.2. It demonstrates how the PSK of a WPA or WPA2 protected network can be retrieved by capturing a 4-way handshake and brute forcing the key using a dictionary.

6.2.1 Setup

To be able to perform this attack the target network was configured to use WPA2 in personal mode and a common English dictionary word was chosen as PSK. As dictionary we used a list of about a million common passwords [12]. For the attack to work we also need a legitimate client that at some point connects or reconnects to the network allowing us to capture the 4-way handshake. To accomplish this we simply used one of our own smartphones.

6.2.2 Execution

As previously stated the wireless card needs to be in monitor mode to be able to capture traffic from networks. There are multiple ways of accomplishing this but the *aircrack-ng suite* provides a simple tool called *airmon-zc*. wlan0 is the interface name of the wireless card and after enabling monitor mode *airmon-zc* will create a new interface named wlan0mon.

airmon-zc start wlan0

To be able to attack the target network some information about it is needed. This includes the APs MAC address and what channel it is operating on. The tool *airodump-ng* can be used to capture Wi-Fi communication but it can also be used to scan for the target network.

airodump-ng wlan0mon

Executing this command will display information about all the networks and clients in the vicinity. Once the target network is identified its MAC address and channel can be used to filter out communication of just the target network. By adding a few options to the previous command all the communication of the target network will be captured to a file.

airodump-ng -c 6 --bssid 00:24:01:A9:5F:2C
-w dump wlan0mon

If a client connects to the network at this point the 4-way handshake will be captured. It can however take a while before a client connects or reconnects on their own. To accelerate the process an already connected client can be deauthenticated from the AP by sending disassociate packets to it, forcing it to reauthenticate and allowing the 4-way handshake to be captured. To accomplish this we used the tool *aireplay-ng* while still capturing data with *airodump-ng*. In order to deauthenticate the client its MAC address is needed and this information is revealed by *airodump-ng* in the same way as the APs MAC address was.

aireplay-ng --deauth 5 -a 00:24:01:A9:5F:2C
-c 81:9D:EC:07:C3:5F wlan0mon

After the 4-way handshake is captured no more interaction with the network is needed. We can now use the data from the captured handshake to mount an offline dictionary attack. To do this the tool *aircrack-ng* is used with the dictionary and dump files as parameters.

aircrack-ng -w dictionary.txt dump-01.cap

Aircrack-ng tries the passwords in the dictionary one after the other and if the PSK exists it will find and display it.

6.2.3 Result

Performing the attack was straight forward and we did not have any difficulties completing the attack. In our attack we used the PSK: “inspiration” and with a testing speed of about 1800 keys per second it was recovered in 4 min and 21 seconds. The result from *aircrack-ng* can be seen in figure 4.

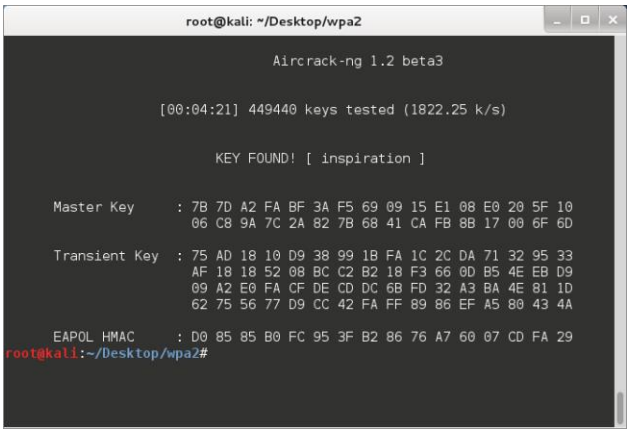

Figure 4: Result from aircrack-ng

When testing keys we only used the CPU which resulted in a relatively low testing speed. In comparison you can achieve speeds several orders of magnitude faster by also utilizing one or multiple GPUs [22]. These speeds are for attacks where you perform all calculations for each tested PSK during the attack. If you use precalculated rainbow tables the attack itself can go even faster. It is of course a tradeoff between time and memory.

With the knowledge of how quickly our attack completed when the PSK was a common English word, you realize how weak the protection is when a bad PSK is used. This shows how important it is to have a long and random PSK so that you are out of reach from brute force and dictionary attacks.

6.3 Attack on WPS

The attack on WPS demonstrated in this section is based on the vulnerabilities discussed earlier in section 5.2. It demonstrates how you can brute force the WPS PIN code to attain network credentials and thereby circumvent the security provided by WPA and WPA2.

6.3.1 Setup

To be able to perform this attack only a target network is needed. The network was configured to have WPS enabled and use WPA2 in personal mode together with a long random string as PSK. This key was chosen to highlight that the complexity of the PSK does not affect the outcome of this attack.

6.3.2 Execution

As in the attack against WPA(2) the wireless card needs to be in monitor mode and it is done in the same way as previously described.

airmon-zc start wlan0

The information needed in order to launch the attack against the target network is the same as in the WPA(2) attack meaning that both the APs MAC address and channel is required. This information can be obtained using *airodump-ng* as in the previous attack but it can also be done with a tool named *wash* that is specifically designed to identify WPS enabled APs.

wash -i wlan0mon

Using this tool also allows you to reveal more information specific to WPS such as what version of WPS the AP supports and if the WPS function is locked or not.

Once the MAC address and channel is known *reaver* can be used to launch an attack against the AP. *Reaver* allows you to configure many parts of its operation using flags. Information about what they do and how they are used can be explained by using the "--help" flag. For our attack we used the "S" flag to speed up the attack by using small Diffie-Hellman secret numbers and the "v" flag to output verbose information about what keys were tested.

reaver -i wlan0mon -c 6 -b 00:24:01:A9:5F:2C -S -v

During execution *reaver* brute forces every possible PIN starting with the first half followed by the second and once the complete PIN is found it will recover and display the PSK.

6.3.3 Result

Performing the actual attack was rather easy once we had acquired the right equipment. After we encountered problems during our first attempts, we learned that *reaver* is very picky when it comes to compatibility with wireless cards and drivers. But once we had bought a new compatible card there were no further problems.

In our attack the default PIN of the AP was set to "14587252" and with a testing speed of about 3 seconds per PIN the attack finished after 1 hour and 50 minutes. The end result can be seen in figure 5.

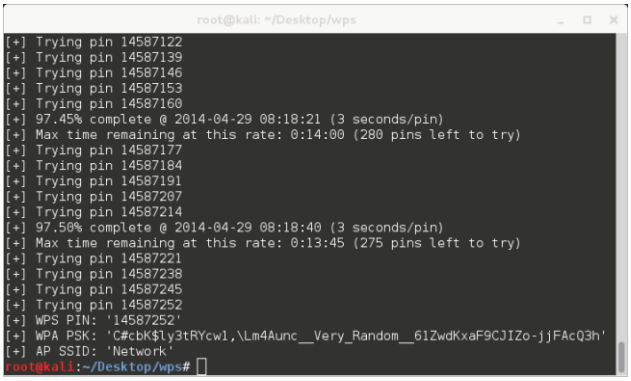

Figure 5: Result from reaver

Since *reaver* tests the PINs in a chronological order our attack was finished relatively fast compared to the average time for this speed which would have been about 4 hour and 35 minutes. Unlike the WPA(2) attack, the speed at which *reaver* can perform tests is not limited by the attackers computing power. Instead it depends on the speed at which the AP can process WPS requests and if there are any mitigations in the form of lockdown periods. In our case the AP did not have any lockdowns and we were therefore solely limited by the APs performance. Even if there had been lockdowns it would not have prevented the attack from eventually succeeding. If the lockdowns had been lengthy and frequent however, they would have severely slowed down the attack. For example a 60 minute lockdown after five failed attempts would have increased the average time to complete our attack to about 46 days.

7. Summary and Conclusions

When WLANs was first introduced the goal was to make their security equivalent to that of a wired network. This effort resulted in WEP which looked great on paper, but it quickly proved to be broken by design. The failure of WEP started an increased effort to develop a secure standard, learning from the previous mistakes. WPA was pushed as an interim solution implementing most of the new techniques in what would later be fully realized as WPA2. When properly configured, WPA2 is today considered to provide sufficient protection, but vulnerabilities in coexisting protocols can still threaten its security as proven by the revelation of the flaws in WPS.

In this report we have reviewed the progression of WLAN security protocols with a focus on vulnerabilities that allows for key recovery attacks. For each protocol we have given a description of the security features they provide and enough details to give an understanding of how their flaws can be exploited. We have explained the workings of existing attacks utilizing these flaws to recover the PSK and also successfully performed two of the attacks ourselves.

When preparing for the attacks we had no problems finding the tools and hardware that was needed. Both step by step guides and detailed demonstrations are easily found browsing the internet. This can of course be seen as bad in the sense that almost anyone can easily acquire everything he or she needs to perform an attack. The high availability has however also meant that the security of WLANs has been thoroughly tested and analyzed. To us, the possibility of anyone being able to inspect and evaluate techniques is a good thing, because it increases the probability of security flaws being detected which in turn leads to the development of more robust protocols.

Even though WPA2 is considered secure, much of its security still depends on the end user knowing how to configure the network correctly. Many home users do not bother to check what encryption is used on their APs or have any understanding of what constitutes a strong or weak password. Unawareness of this can lead to many networks in practice having poor security even though they use WPA2. WPS tried to solve this problem by making it easier for users with little knowledge about network security to configure their networks, but unfortunately it introduced more vulnerabilities leading to an even worse situation. This showcases the difficulty of predicting how changes and additions to a system can affect its security. In the end the only way to stay secure as a user is to have knowledge about good practices and keep up to date with security threats that may affect you.

References

[1] IEEE 802.11-2012. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. (2012 revision). IEEE-SA. 6 February 2012.

[2] Borisov, N., Goldberg, I., & Wagner, D. (2001, July). Intercepting mobile communications: the insecurity of 802.11. In Proceedings of the 7th annual international conference on Mobile computing and networking (pp. 180-189). ACM. [doi:10.1145/381677.381695]

[3] Wong, S. (2003). The evolution of wireless security in 802.11 networks. In WEP, WPA and 802.11 standards. GSEC Practical v1. 4b.

[4] Fluhrer, S., Mantin, I., & Shamir, A. (2001, January). Weaknesses in the key scheduling algorithm of RC4. In Selected areas in cryptography (pp. 1-24). Springer Berlin Heidelberg. [doi:10.1007/3-540-45537-X_1]

[5] Klein, A. (2008). Attacks on the RC4 stream cipher. Designs, Codes and Cryptography, 48(3), 269-286. [doi:10.1007/s10623-008-9206-6]

[6] Tews, E., Weinmann, R. P., & Pyshkin, A. (2007). Breaking 104 bit WEP in less than 60 seconds. In Information Security Applications (pp. 188-202). Springer Berlin Heidelberg. [doi:10.1007/978-3-540-77535-5_14]

[7] Bulbul, H. I., Batmaz, I., & Ozel, M. (2008, January). Wireless network security: Comparison of WEP (wired equivalent privacy) mechanism, WPA (wi-fi protected access) and RSN (robust security network) security protocols. In Proceedings of the 1st international conference on Forensic applications and techniques in telecommunications, information, and multimedia and workshop (p. 9). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[8] Aircrack-ng.org (2014). Aircrack-ng tool documentation. Available: http://www.aircrack-ng.org/doku.php?id=aircrack-ng . Last accessed 30 April 2014.

[9] Liu, Y., Jin, Z., & Wang, Y. (2010, September). Survey on security scheme and attacking methods of WPA/WPA2. In Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th

International Conference on (pp. 1-4). IEEE. [doi:10.1109/WICOM.2010.5601275]
[10] Kumkar, V., Tiwari, A., Tiwari, P., Gupta, A., & Shrawne, S. (2012). Vulnerabilities of Wireless Security protocols (WEP and WPA2). International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 1(2), 34-38.
[11] Sakib, A. N., Jaigirdar, F. T., Munim, M., & Akter, A. (2011). Security Improvement of WPA 2 (Wi-Fi Protected Access 2). International Journal of Engineering Science and Technology, 3(1).
[12] RenderLab.net (2007). Church of Wifi WPA-PSK Lookup Tables. Available: http://www.renderlab.net/projects/WPA-tables/ . Last accessed 30 April 2014.
[13] Tews, E., & Martin B. (2009). Practical attacks against WEP and WPA. Proceedings of the second ACM conference on Wireless network security. ACM, 2009. [doi:10.1145/1514274.1514286]
[14] AirTight Networks (2010). WPA2 Hole196 Vulnerability. Available: http://www.airtightnetworks.com/WPA2-Hole196 . Last accessed 30 April 2014.
[15] Todo, Y., Ozawa, Y., Ohigashi, T., & Morii, M. (2012). Falsification Attacks against WPA-TKIP in a realistic Environment. IEICE TRANSACTIONS on Information and Systems, 95(2), 588-595.
[16] Vanhoef, M., & Piessens, F. (2013, May). Practical verification of WPA-TKIP vulnerabilities. In Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security (pp. 427-436). ACM. [doi:10.1145/2484313.2484368]
[17] Viehböck, S. (2011). Brute forcing wi-fi protected setup. Available:http://sviehb.files.wordpress.com/2011/12/viehboeck_wps.pdf . Last accessed 30 April 2014.
[18] Tactical Network Solutions. (2011). Cracking WiFi Protected Setup with Reaver. Available: http://www.tacnetsol.com/news/2011/12/28/cracking-wifi-protected-setup-with-reaver.html . Last accessed 30 April 2014.
[19] Kali Linux (2014). Penetration testing distribution. Available: http://www.kali.org/ . Last accessed 30 April 2014.
[20] Aircrack-ng.org (2014). Aircrack-ng Suite. Available: http://www.aircrack-ng.org . Last accessed 30 April 2014.
[21] Tactical Network Solutions (2013). Reaver. Available: https://code.google.com/p/reaver-wps/ . Last accessed 30 April 2014.
[22] Hashcat.net (2014). Oclhashcat password cracker. Available: https://hashcat.net/oclhashcat/ . Last accessed 30 April 2014.
[23] Kaliski, B. (2000). PKCS# 5: Password-based cryptography specification version 2.0. Available: http://tools.ietf.org/html/rfc2898