Practical WLAN security, spring 2008

Gustav Nykvist, Johannes Larsson
Email: {gusny326,johla723}@student.liu.se
Supervisor: David Byers, {davby@ida.liu.se}
Project Report for Information Security Course
Linköping University, Sweden

Abstract

The report is about security issues related to wireless area networks. First the report covers some basic knowledge concerning the wireless standards and then known flaws related to them. We have, during the project, tested already existing tools that exploits the vulnerabilities in WEP and WPA. It also covers, how the tools we used work, but not in full detail. The first attack that was tested was obtaining the secret WEP key through cryptanalysis. To attack WPA we used a brute force attack. Finally, we mention what people should think of when setting up a WLAN, to be relatively sure that no one will be able to intrude on their network.

1. Introduction

WLANs, wireless local area networks, have increased in popularity. Today wireless networks are commonly used in private households. Over the years there have been issues with the security related to wireless networks, and many of the attacks are not that hard to perform. Existing scripts for attacking wireless networks are available for everyone to download from the Internet. Since many of the attacks do not need anything more than off-the-shelf equipment, every person with moderate computer knowledge is made a possible attacker.

The goal with this paper is to show how easy it is for a script-kiddie, a malicious hacker with no knowledge of creating scripts by him/herself, to perform a serious attack on a WLAN that is using the security standard WEP. This will be shown by performing a couple of attacks with the help of already existing scripts. We also strive to find out and point out what a WLAN home user should think of when setting up a private WLAN from a security perspective. Finally, if time allows, we want to program and perform our own DoS, denial of service, attack against a wireless network that is using WEP.

2. Background

The background is divided into two subsections; Wireless Standards and Wireless Security Standards. The first covers the basics of a wireless local area network and the second the security standards.

2.1 Wireless Standards (802.11)

Within a wireless local area network several different standards exists. These standards are developed by IEEE [1]. The British Standards Institution [2] definition of a standard is, “a standard is an agreed, repeatable way of doing something”. The set of standards this paper will deal with is the one used for computer communication, IEEE 802.11.

In 1997, the original version of 802.11 was released. It was operating at 2.4 Ghz and at a data rate of 2Mbit/s. Later on the 802.11a and 802.11b was both released, in 1999. They differed in most aspects: data rate, throughput, range, frequency. In the favor of 802.11a; it had a higher promised data rate, it was operating on a less used frequency spectrum (5 Ghz). On the other hand signals at a higher frequency are more readily absorbed [3]. Hence its range was shorter. The 802.11b was operating on the crowded 2.4 Ghz band, in which microwaves and cordless phones also are active. However 802.11b became the standard since it was an upgrade of the original version and cheaper than 802.11a. In 2003 the 802.11g was released, providing high speed (up to 54 Mbit/s) in the 2.4 Ghz spectrum. The hardware used in 802.11g is backward compatible with the one used in 802.11b. The next standard is 802.11n which promises larger range and six times as good speed as 802.11g. The new standard hasn’t been fully deployed yet. However products implementing drafts of the new standard are available in stores right now, one example is the D-LINK DIR-655 that got the following standards: 802.11b/g/n-draft v2.0.

In a WLAN you need something that’s called MAC, medium access control, to provide control over the medium, which in a WLAN is the radio spectrum. The main reason for using MAC is to the decrease the number of collisions, resulting in a higher throughput. The standard MAC in 802.11 is CSMA/CA, carrier sense multiple access with collision avoidance. It's basically about listening if anyone is sending, if so wait a random amount of time and check again, else just start sending. Another option is to use different types of frames, RTS (request to send) and CTS (clear to send).

As David Byers [4] mentioned in his slides, "Security was a top concern when IEEE 802.11 was defined. The standard includes an optional protocol called WEP that is designed to provide the same level of security as a wired network". More information about WEP and the other standards will be covered later on in the paper.

2.1.1 Architecture

All components able to communicate over the wireless medium are called stations. They all got a unique id called MAC address. A typical example of a station is a mobile device, such as a laptop. The principal component of a WLAN is the basic service set, BSS and the access point, AP. A BSS is the service that an AP provides, the AP itself is a station that is wired, often to the Internet. A WLAN that deploys access points is referred to as an infrastructure network. But of course there are WLANs without any access points, they are called ad-hoc networks. The idea is to connect two or more stations as they need to communicate, forming a network of their own, but with no connection to the outside world. No Internet.

In an infrastructure network stations need to associate to the AP before they can start sending or receiving data. This requires that the access point is configured with a name, also referred to as Service Set Identifier, SSID. These names are viewable when browsing wireless networks on a regular computer with a wireless network interface card, NIC. When a station is associating it can be described as creating a virtual wire between itself and the AP. In 802.11 access points periodically sends announcements, also referred to as beacon frames, including the APs SSID and MAC address. Clients listen for these announcements to discover access points, it's called passive scanning. But they can also probe for them by broadcasting a request, this is referred to as active scanning. When the client is done it unplug the virtual wire by disassociating.

2.1.2 Frame

Frames are data packets of a fixed size encoded by the link layer protocol, in this case 802.11. Every frame got a control field that consists of much information: the version of the 802.11 protocol, the type of the frame, a more fragment field, if WEP is enabled or not and many more. All frames also got four address fields, three of them are needed to send a network-layer datagram to an AP and onwards to a router interface. The last one is used in ad hoc mode. There are also a frame sequence field to avoid receiving duplicates and a duration field which describes how long you want to reserve a channel. Reserving channels is used when sending data, RTS (request to send) and CTS (clear to send) frames. These frames accompanied with the ACK (acknowledge) frame is all referred to as control frames.

Another type of a frame is the management frames. They are used in order to establish and maintain communication. Different types of them are:

- Authentication frame, are used in the context of access points deciding if to accept or reject setting up communication. In an open system the connecting station just sends one authentication frame and the AP answers by sending one back, declaring if it accepts or rejects.

  As opposite to the open system there are those with a shared key. Here the AP initializes a challenge as its first step after it has received an authentication frame from the station. It then has to encrypt it with its shared key and send it back to the AP. The access point then encrypts it and compares the result with the correct key it got stored. Depending on the result it decides if to accept or reject.
- De-authentication frame, is the opposite, used when to terminate a communication.
- Association frame requests, enables synchronization and helps allocating resources for the station, that's sending it to the AP.
- Disassociation frames, are also sent by the station. It is the nice way to end an association, as a result the access point can relinquish resources that was previously used by the station.
- Beacon frames are an announcement of the existence of the AP. These are broadcasted periodically. A broadcast is referred to as sending to all, in this context it will only be those that are within range. The frame consists of the SSID of the AP and some other parameters.
- Probe requests are sent by the station to find access points it is familiar with.

At last, there is one last type, the data frame. It's used when transporting data. [10]

2.2 Wireless Security Standards (802.11i)

This section covers the three Security standards briefly.

2.2.1 WEP

Wireless equivalent privacy protects the link-level data during transmission and is used in 802.11 networks. The WEP security goals are to enforce confidentiality, access control and data integrity. Confidentiality refers to protecting the communication from eavesdropping. Access control refers to protecting the access to the network and data integrity refers to protecting from tampering on transmitted packages. To achieve the security goals the protocol makes use of different security mechanisms, which will be described briefly. Checksumming, first an integrity checksum c(M) is computed on the message M. After that the two are concatenated to obtain the plaintext P=<M,c(M)>. This will be used as the input for the second stage. During the second stage, encryption, the plaintext P is encrypted using the algorithm RC4 (which is implemented in a non-standard way). An initiation vector (IV) v is chosen. RC4 generates, as a function of the pre-shared key k and the IV v, a keystream. A keystream is in example a sequence of pseudorandom bytes. The keystream is denoted RC4(v,k). After this the plaintext is added with exclusive-or to the keystream to obtain the ciphertext. This is denoted C=XOR(P,RC4(v,k)). Finally, the IV and the ciphertext are transmitted over the radio link.

A → B: v,(XOR(P,RC4(v,k)) where P=<M,c(M)>, is a symbolical representation of the checksumming, encryption and transmission. The recipient simply reversers the encryption process when he wants to decrypt a frame protected by WEP. Symbolically this can be written P = XOR(C,RC4(v,k) = XOR(XOR(P,RC4(v,k)),RC4(v,k) = P. To verify the checksum on the decrypted plaintext P the recipient splits it into the form the form <M,c>, re-computes the checksum c(M), and checks that it matches the received checksum c [5].

2.2.2 WPA

WiFi protected access, WPA, has the same purpose as WEP. To meet some of the security issues in WEP, WPA uses WEP but with a new front-end called TKIP, temporal key integrity protocol. WPA uses the RC4 stream cipher algorithm, the same that is used in WEP. The intention with TKIP was that it should be an interim solution, and that it should work with already deployed hardware. This imposes some constraints. Allow deployed systems software or firmware to be upgradeable. Current WEP hardware implementation should be allowed to remain unchanged. Finally, minimize performance degradation imposed by the fixes. To address the known flaws in the WEP protocol TKIP uses a set of algorithms. To defeat forgeries there is a message integrity code, MIC, referred to as Michael. To defeat replay attacks there is a packet sequencing discipline. And to prevent cryptanalysis attacks there is a per-packet key mixing function. The MIC, packet sequencing and per-packet key mixing will be described briefly. The Michael algorithm calculates the keyed function and then sends the result as a tag to the data to the receiver. The receiver re-computes the value and compares the values. If the values match the receiver assumes that the data is authentic and if they don't the receiver rejects the data. Michael partitions a 64-bit key into 32-bit blocks. After this, shifts and exclusive ORs are used to process the two 32-bit blocks in to two 32-bit registers. The two 32-bit registers will represent the result, a 64-bit authentication tag. The level of security of the MIC is measured in bits.

TKIP also mandates countermeasures because Michael is too weak to stand alone. If a MIC validation error is discovered TKIP requires a rekey. There is a maximum limit for rekeying, once per minute.

Packet sequencing is used to protect against replay attacks. WEP is extended by TKIP to use a 48-bit sequence number. However, because of existing implementation constraints it associates the sequence number with the encryption key instead of the MIC key. TKIP then mixes the sequence number into the into the encryption key, and encrypts the WEP ICV and the MIC. A replay attack will be translated into ICV or MIC failures. Per packet key-mixing is used to defend against cryptanalysis attacks. TKIP uses a new per-packet encryption key construction that is based on a mixing function. The base key, transmitter MAC address and packet sequence number are the inputs to the mixing function and the output is a new packet WEP key [6].

2.2.3 802.11i / WPA2 / RSN

A long awaited security standard for wireless networks, designed without any restriction of being backwards compatible with old hardware. This is announced to be a long-term solution. Instead of using RC4 like WEP and WPA do, it uses AES in counter mode to make the block cipher work as a stream cipher. This assures 802.11i to provide confidentiality, integrity and data origin. It also got:

- Longer keys than WEP do
- A new integrity check
- Replay protection

WPA2 can run in two different modes, one for home and small offices, this referred to as WPA-PSK, pre-shared key mode. When running in this mode there is no need for an 802.1X authentication server. The only thing a user must supply to access the network is a passphrase, these consist of 8 to 63 ASCII characters. It's also possible to use a passphrase of 64 hexadecimal numbers. These passphrases must be known by the access point and are stored on them. They are also usually stored on the computer in the operating system to increase the usability, no one wants to re-enter the passphrase each time your station re-associates.

The other mode WPA2 can run in is the Enterprise mode. This mode uses an 802.1X authentication server. The 802.1X ensures that a station must be authenticated, in order to gain access to other LAN resources. As the Wi-Fi Alliance mentions "Hackers can break encryption codes by intercepting and analyzing large amounts of data, but breaking codes takes time" [9]. This can be prevented by changing codes every five minutes or so, making sure that the codes the hacker breaks are useless. The 802.1X changes the codes automatically.

3. Security

This section will cover the security aspects.

3.1 Flaws in Wireless Standards

One problem here is that all peripherals operating on the same frequency as the WLAN will interfere it, as a result compromising the availability. As mentioned before microwaves and cordless telephones operate on the 2.4 Ghz ISM band, and by using these you will suffer from lower or no performance on your WiFi network. There are many guides on the Internet describing how to build your own Radio Frequency (RF) Jamming device.

Another problem is the range, which is specified indoors with regular antennas. So by using larger and more expensive antennas you can increase the range significantly. One example is using a WiFi yagi rifle which can increase the range from the specified 100 m to 16 km [4]. This proves the point that you never should use the range as security measure.

3.1.1 Architecture

Since all stations got a unique id, the MAC address, it should be very hard to falsify who you are. However it's not. Today it's possible and easy to change the MAC address on most hardware, also known as MAC spoofing.

3.1.2 Frames

Within the security standards we are covering in this paper the only frame that is protected is the data frame. Hence both the control and management frame are unprotected, which in fact is the same as being transmitted in the clear.

3.2 Flaws in Wireless Security Standards

The standards and the problems related to them.

3.2.1 WEP

WEP fails to meet all of the three security goals. Since WEP uses a single pre-shared key. There is often a device that stores the key, and if that device is compromised or lost the key must be changed. Since there is no key management protocol in WEP, changing key can take a lot of time. However, this kind of problem is often ignored. The threat that is considered as the most serious in practice is that the key can be retrieved through cryptanalysis. WEP uses the RC4 algorithm, but it is implemented in a non-standard way. The RC4 per-packet key is created by concatenating a base key with a 24-bit per-packet nonce, called the WEP IV. If an eavesdropper can obtain several million encrypted packets where the first part of the plaintext is known it is possible for the eavesdropper to deduce the RC4 key. This is possible by exploiting the properties of the key schedule. Since the attack only requires off-the-shelf software and hardware and is purely passive it is a dangerous threat.

Since the cryptanalysis attack perhaps is the most serious threat against WEP, it is worth mentioning some history behind it and briefly how it works. Fluhrer, Mantin and Shamir published an article in 2001 concerning the RC4 algorithm that is implemented in WEP and a possible way to break it [17]. They did not implement the attack themselves, but it was accomplished soon after by Stubblefield, Ioannidis and Rubin [18]. The attack, known as FMS, requires about 4 000 000 – 6 000 000 captured data packets to recover the WEP key. A hacker named KoReK improved the FMS attack in 2004 and the required data packets to obtain a 104 bit WEP key was reduced to about 500 000 – 2 000 000. In 2005 another analysis of the RC4 stream cipher was presented by Andreas Klein. He showed that there were even more correlations between the RC4 key stream and the key than Fluhrer, Mantin and Shamir had found. And that those correlations could be used to break WEP, when used in WEP like usage mode. Twes, Weinmann and Pyshkin managed to extend and optimize Klein's attack for usage against WEP. Using this attack you only need 40 000 captured packets to get a 50% probability to recover a 104 bit WEP key. The success probability is increased to 80% with 60 000 captured packets and 95% for 85 000 captured packets. When targeting a 40 bit key the same attack can be used with even higher probability of success [19].

To increase the understanding of the cryptanalysis attack against WEP, the weaknesses found by Fluhrer, Mantin and Shamir will be described very briefly. The weaknesses can be found in the Key Scheduling Algorithm (KSA) and how it derives the initial state from a key of variable size. The first weakness is the fact that it exist large classes of weak keys. In the weak keys a small part of the keys determines a large number of bits of the initial permutation outputs (KSA outputs). The second weakness applies when a part of the key is exposed to the attacker. The attacker can re-derive the secret part by analyzing the initial word of the key streams, whilst exposing it against numerous different exposed values [17]. To understand how this really works further reading is advised.

All security is lost when the eavesdropper have obtained the WEP key. The problems with the WEP design are as follows, the 24-bit IVs are too short. The CRC checksum, used for integrity protection, is insecure. The key and IV combining enables cryptanalysis attack. Finally there is no integrity protection provided for source and destination addresses. All this results in that confidentiality is put at risk. There is no prevention for adversarial modification of intercepted packets. And finally, passive eavesdropping can obtain the key, by only observing encrypted packets [6].

3.2.2 WPA

WPA is based on WEP, but has TKIP as a frontend, and in difference to WEP it actually provides some security. However, distributing the pre-shared key is still a bit of a problem. Management frames in WPA are, just as in WEP, unprotected. This is a weakness since attackers can spoof those frames and do harm to the network [4]. Another problem with WPA is weak passphrases. If the WPA key is chosen from a regular word, it can probably be found within a dictionary. As a result an attacker can do a pre-compiled dictionary attack. To do this the attacker passively intercepts key exchange messages. The key exchange message sent between an access point and a station occurs only in the beginning of a connection. But with the help of dissociate messages the attacker can force the key exchange messages to be resent. It only takes a few minutes for the attacker to obtain the key and get access to the network.

There is a method that have done dictionary attacks practical to use against WPA-PSK, it is called Rainbow Tables. The idea behind Rainbow Tables is that you do a brute force attack once, and then use the result to accelerate the attack the next time you want to crack a hash. Rainbow Tables use a reduction function that maps hashes to plaintext, it does the reverse of a hash function but not the inverse. By using this you can represent millions of hashes with only one single starting plaintext and one single finishing hash [20]. Let's call this a chain. The starting point and ending point of the chain are stored in a table, and when you want to crack a hash you simply regenerate all the hashes with the help of the values in the table. To speed up the process you use several chains, it is a trade-off between memory and time [21]. Rainbow Tables differ from it's predecessors in the approach to solving the problem of certain plaintexts never being reduced to. The predecessors tried to solve this by using several small tables with different reduction functions. Rainbow Tables only use one table, where every column in the table has a different reduction function. However, it is still unlikely that all the plaintext in the desired set will be hashed. But for a given number of chains the chances are higher. This also reduces the chance of chain merges and solves the problem with loops [20].

A solution to the problem described above exists, to use keys that are longer than 20 characters and only contains gibberish. However, those keys are often hard to remember and that is a problem, when WPA is used in pre-shared key mode [7]. Today it is often common that the secure WPA key is written down on, perhaps underneath, the access point. This solves the problem, but writing down passwords does impose a risk to the network.

3.2.3 802.11i / WPA2 / RSN

The weakest link here is still the passphrase used in home mode. Since home users rarely got the knowledge or resources to have an 802.1X server running. A dictionary attack could probably defeat most common passwords very quick.

Another problem is that the protection of the management frames in 802.11i is still omitted. As a result, it is like the older security standards, vulnerable to denial of service attacks. One example is to send forged Disassociate frames, which are a type of management frame. These could be sent both to a single user or broadcasted on the behalf of the AP. As a result the affected user(s) would be disconnected, as long as the attack continues. One solution to this issue is the new 802.11w, which is the upcoming security standard that comes with protected management frames [11]. This eliminates the attacks based on forged management frames. But there are still one frame type unprotected, the control frame. According to Joshua Wright the task group working on 802.11w have no intention to protect the control frames [13]. Since they are unprotected it's possible to exploit the RTS, CTS and ACK frame to dominate the medium [12]. This type of attack is also a denial of service attack.

4. Practical Work

This section is divided in a chapter about the preparations before attacking, what tools there are out there and what attacks we tried out.

4.1 Preparations

The preparations phase consists of a couple of tasks, which are needed to be completed, in order to use the available tools.

4.1.1 The driver issue

The project specification describes that the attacks should be performed on a Linux laptop. There are several reasons for not doing the attacks from a Microsoft Windows laptop. One problem is that drivers created for the Microsoft Windows platform are developed by a third party. The source code of these drivers is not available publicly. Hence there is little or no support from the ones writing the tools [8]. There is more to read about the tools in section 4.2.

The device drivers available in Linux are divided in two different types, depending if the manufacture have released the specifications. If they have, there is a good chance that there is an open source project writing drivers. If not, the developers need to reverse engineer the hardware to be able to understand how the device works, in example how it communicates over a certain bus. The support for Linux by vendors, referred to as third party, is in most cases limited. A popular card has a greater chance of being compatible, as a result of having more people willing to develop, debug and maintain the driver.

4.1.2 The search of a compatible card

Whilst following a tutorial we quite soon realized that the hardware, wireless network interface card, we were using wasn't supported. It was only able to listen to traffic, referred to as be put in monitor mode. The only available drivers out there, was reversed engineered. Not good enough. To fully unleash the power of the tools out there we would need a card able to inject packets.

The idea behind injecting packets, in this context, is to create spoofed packets, on a network that you don't belong to. If you are not able to spoof packets, the collection of IVs when breaking WEP, will be a time consuming process. You just have to wait until you have listened on enough traffic. If you on the other hand can inject your own packets it will go a lot quicker. The art of injecting packets is not only useful when breaking WEP.

With the help of the compatible list at aircrack-ng.org and offensive-security.com we found a wireless NIC which was supported [14] [15]. Now that we had found some possible candidates we needed to match them against what was possible to order from online stores in Sweden. A task easily completed by a major price comparison site. Two days later our new hardware had arrived. Once installed, we had to patch the drivers. Now finally, we were able to inject packets.

4.1.3 Installing the Linux distribution of our choice

The Linux distribution we installed first was Ubuntu, a very user friendly distribution that is known for just working. From Ubuntu you can manually download the tools you want or use the apt-get to install what you want. The later of them is the automatic way that often concludes in a working tool.

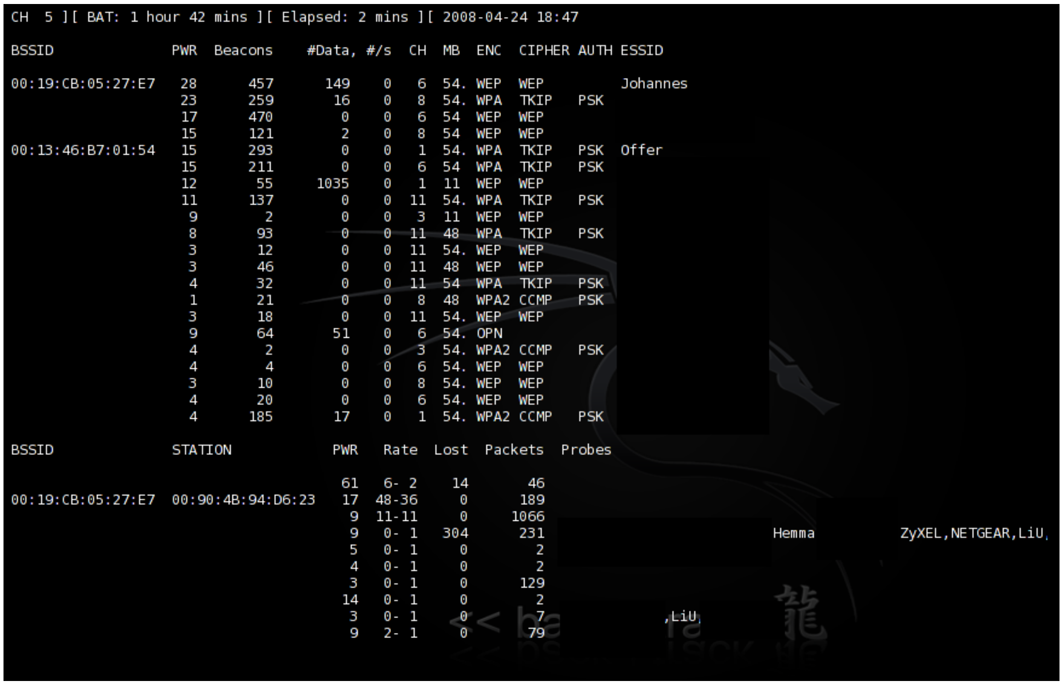
The other distribution we tried out is Backtrack. Since it runs from a CD or USB memory it would be wrong to say that we installed it. But we tried the installation that is included in the Backtrack Beta 3. It's not recommended by the developers and we are not so pleased with it either, since it corrupted the master boot record on one of our laptops. The distribution is focused on penetration testing, and because of all useful tools it provides it has become popular distribution among hackers [16].

4.2 Common scripts and tools

There are a lot of tools that can be used to analyze and to attack WLANs. In this chapter some of them will be mentioned briefly.

Aircrack-ng is a program for cracking WEP and WPA-PSK. Since Aircrack-ng will be used during the practical part of the project it will be explained in more detail. The Aircrack-ng suite contains the following modules: aircrack-ng, airdecap-ng, aireplay-ng, airmon-ng, airodump-ng, airtun-ng and packetforge-ng. The different modules serve different purposes and all are not needed when trying to obtain a WEP key. In fact you can do several different kinds of attacks with the help of the Aircrack-ng suite. Some attacks that you are able to perform are the following: de-authentication, fake authentication, interactive packet replay, ARP request replay attack, KoReK chopchop attack, fragmentation attack, injection test. The de-authentication attack forces connected clients de-authenticate. This is accomplish through sending spoofed disassociate messages. There are several reasons why you might want to perform this kind of attack. You can recover a hidden ESSID, capturing WPA/WPA2 handshakes and generate ARP requests. You can also use de-authentication to perform a denial of service attack by flooding the network with disassociate messages. Fake authentication allows the attacker to perform two kinds of WEP authentication, open system and shared key, and associate with the access point. The reason why an attacker performs this is because he or she wants to obtain an associated MAC address, which can be useful in various attacks. It is not possible to use these attacks against WPA/WPA2 access points. Interactive packet replay is when the attacker is injecting packets on the network. This is useful when trying to obtain initialization vectors (IV) at a higher rate. ARP replay attack is also used by the attacker to obtain IVs at a higher rate. The idea is to capture an ARP packet and then replay it to the AP which makes the AP to replay with a new ARP that contains a new IV. The KoReK chopchop attack can when successful determine the plaintext for a WEP data packet without knowing the key. The fragmentation attack is helpful when trying to achieve the pseudo random generator algorithm (PRGA).

This information can be useful in different injection attacks. Finally, the injection test is useful when testing what APs in the area that responds to probes, and it also lists the connection quality to those APs. The injection test is also used to make sure that the network interface card (NIC) is able to inject packets. [22]



```
CH  5 ][ BAT: 1 hour 42 mins ][ Elapsed: 2 mins ][ 2008-04-24 18:47

BSSID              PWR  Beacons    #Data, #/s  CH  MB  ENC  CIPHER AUTH ESSID

00:19:CB:05:27:E7   28      457      149    0   6  54. WEP  WEP         Johannes
                    23      259       16    0   8  54. WPA  TKIP   PSK
                    17      470        0    0   6  54  WEP  WEP
                    15      121        2    0   8  54  WEP  WEP
00:13:46:B7:01:54   15      293        0    0   1  54. WPA  TKIP   PSK  Offer
                    15      211        0    0   6  54  WPA  TKIP   PSK  [redacted]
                    12       55     1035    0   1  11  WEP  WEP
                    11      137        0    0  11  54. WPA  TKIP   PSK
                     9        2        0    0   3  11  WEP  WEP
                     8       93        0    0  11  48  WPA  TKIP   PSK
                     3       12        0    0  11  54. WEP  WEP
                     3       46        0    0  11  48  WEP  WEP
                     4       32        0    0  11  54  WPA  TKIP   PSK
                     1       21        0    0   8  48  WPA2 CCMP   PSK
                     3       18        0    0  11  54. WEP  WEP
                     9       64       51    0   6  54. OPN
                     4        2        0    0   3  54. WPA2 CCMP   PSK
                     4        4        0    0   6  54. WEP  WEP
                     3       10        0    0   8  54. WEP  WEP
                     4       20        0    0   6  54. WEP  WEP
                     4      185       17    0   1  54. WPA2 CCMP   PSK

BSSID              STATION            PWR   Rate  Lost  Packets  Probes

                                       61   6- 2    14       46
00:19:CB:05:27:E7  00:90:4B:94:D6:23   17  48-36     0      189
                                        9  11-11     0     1066
                                        9   0- 1   304      231  [redacted]           Hemma          ZyXEL,NETGEAR,LiU,
                                        5   0- 1     0        2
                                        4   0- 1     0        2
                                        3   0- 1     0      129
                                       14   0- 1     0        2
                                        3   0- 1     0        7  [redacted]  ,LiU,
                                        9   2- 1     0       79
```

Figure 1. A typical overview from airodump-ng. No filtering

It might be a bit confusing how to combine those modules to perform a successful attack against a WLAN using WEP. This is not covered in this article, but there are a lot of tutorials on the Internet.

Another program that is used for attacking and achieving keys from WLANs is AirSnort. [23]

coWPAtty is a program which is useful when carrying out a dictionary attack. Vulnerable targets for this attack are those WLANs running WPA as a security standard. There are a lot of free dictionaries that are available on the Internet. Before you are able to perform a dictionary attack with coWPAtty you need to capture a TKIP four way handshake between the AP you want to attack and a client. Handshakes take place when a client and AP are setting up a connection [24].

Airsnarf is a script that is useful when setting up a rogue access point. This can be useful when trying to steal usernames and passwords from a public wireless hotspot. The idea is to set up an identical access point as the one used at the public hotspot, and fooling the users to give their login information to the wrong AP [25].

Wireshark is a popular program for network analysis. It runs in Windows, OS X, Linux, Solaris, FreeBSD, NetBSD and other platforms. Wireshark implements a lot of different features, and is not only used to analyze WLAN traffic but network traffic in general [26].

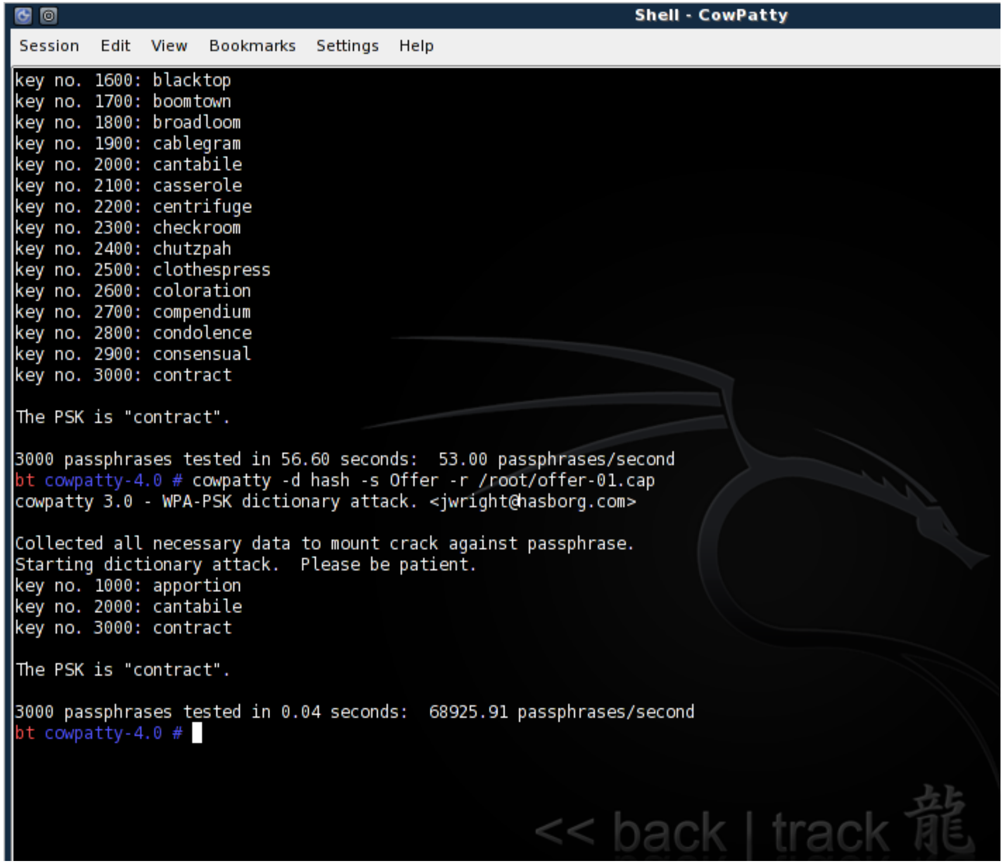
4.3 The attacks we tried

This section will grasp over the attacks we tried out, with some comments on the overall result.

4.3.1 Breaking WEP

With the help of the Aircrack-ng suite we did break WEP, both with and without clients. The first one we tried was attacking an AP which did not filter MAC addresses and had a client authenticated with it. The first thing we did was putting the NIC in monitor mode and starting airodump-ng to sniff traffic and store the information in a file.

After that we used fake authentication to gain an associated MAC address. This was followed by the de-authentication attack and arp reply attack, which was performed simultaneously. When an arp was obtained the reply attack started, replying arps to gain IVs. After this aircrack-ng used the captured data packets for its cryptanalysis attack and obtained the WEP key. The next attack we performed was against an AP that did not filter MAC addresses and with no clients connected to it. This was a bit trickier, the problem here is that there is no client that you can de-authenticate to obtain an arp. To solve this we used the fragmentation attack to receive a PRGA and with that information we could create an arp of our own with packetforge-ng. By injecting the arp we soon got enough IVs to obtain the WEP key.



```
key no. 1600: blacktop
key no. 1700: boomtown
key no. 1800: broadloom
key no. 1900: cablegram
key no. 2000: cantabile
key no. 2100: casserole
key no. 2200: centrifuge
key no. 2300: checkroom
key no. 2400: chutzpah
key no. 2500: clothespress
key no. 2600: coloration
key no. 2700: compendium
key no. 2800: condolence
key no. 2900: consensual
key no. 3000: contract

The PSK is "contract".

3000 passphrases tested in 56.60 seconds:  53.00 passphrases/second
bt cowpatty-4.0 # cowpatty -d hash -s Offer -r /root/offer-01.cap
cowpatty 3.0 - WPA-PSK dictionary attack. <jwright@hasborg.com>

Collected all necessary data to mount crack against passphrase.
Starting dictionary attack.  Please be patient.
key no. 1000: apportion
key no. 2000: cantabile
key no. 3000: contract

The PSK is "contract".

3000 passphrases tested in 0.04 seconds:  68925.91 passphrases/second
bt cowpatty-4.0 # 
```

Figure 2. coWPatty, top = dictionary - bottom = hash

We also performed a denial of service attack by sending forged disassociate packets, both to a single station and to an AP. A client experience such attack will, depending on the cards injection speed, have little or no connection at all to the access point.

4.3.2 Breaking WPA

coWPatty is the tool we decided to try in order to break our victim access point, Offer. We configured it with a password ‘contract’, a regular English word. We started out by de-authenticating the connected client(s). The purpose of it was to receive a TKIP four-way handshake. In order to receive the whole four-way handshake we realized that a good signal is required, else you’ll probably miss some parts of it. When we got it we just ran coWPatty against a dictionary file and a pre-computed hash. There is a big difference in time between using a dictionary file and a pre-computed hash, check Figure 2. The reason is that you don’t need to hash the word and the SSID before comparing when running with a pre-computed hash list, since it already have been done. The hash can be created with a program called genpmk. When you create it, you apply the dictionary file and the SSID of the AP. As mentioned before a lot of pre-computed hash files with regular SSID, like linksys, are available for download. These pre-computed hashes are similar to how Rainbow tables work [24].

5. Conclusions

Hacking WEP is not something that is especially hard. Everything you need can be bought from most retail stores and the attacks are automated and can be downloaded from the Internet. However it can be a bit time consuming, at least it was for us, to get everything up and running. There are many tutorials that can be found on the Internet that describe, step by step, how the attacks should be performed. So no, it is not that hard if you have some basic computer knowledge and a little time and money to spend. This is of course also true when it comes to WPA, however the chance of the attack succeeding is less in comparison to WEP.

WEP is considered to be outdated and should not be used anymore. However it is still commonly used. Only by sniffing the WLAN traffic in the neighborhood where we were working on the project we found out that about 20% of the APs was using WEP. Besides that there were also APs with no protection at all, in this case WEP would be preferred. Bad protection is better than none. If you are able to hack a WLAN you might be able achieve free Internet access, but you might also be able to do far more harm.

One example is using Ettercap [27] to perform a Man in The Middle attack, by arp-cache poisoning. You can use this to perform for example phishing and SSH-downgrading attacks. One of our goals was that we would try to implement our own denial of service attack if time allowed. It's a pity that we have not had the time to do this, since it would give us a far greater understanding relating to the practical part of WLAN security. Script-kiddies are usually not considered to be real hackers, but since there are so many scripts and tools available today they can still do a lot of damage. You can protect yourself against most of their attacks just be being careful and making sure that the equipment and the software you are using are up to date. And since all the tools usually are available to read about and download from the Internet, you can enhance your own knowledge of what kind of attacks script kiddies can perform. Here comes a list of tips for securing your WLAN:

- Turn off the router announcement of its SSID
- Chose an SSID that is not common. To avoid pre-computed hashes
- Pick a good password
- If possibility chose WPA2 as a security measure

Since the Linux version of most of the tools is a command interface one, it can run on a really slow computer. On the other hand it may discourage users whom only got experience from a graphical user interface.

References

[1] Institute of Electrical and Electronics Engineers http://www.ieee.org/ [2008-04-01]

[2] http://www.bsi-global.com/en/Standards-and-Publications/About-standards/What-is-a-standard/ [2008-04-01]

[3] Bradley Mitchel, About.com: Wireless / Networking. http://compnetworking.about.com/cs/wireless80211/a/aa80211standard.htm [2008-04-03]

[4] David Byers, IDA at Linköping University. http://www.ida.liu.se/~TDDD17/lectures/slides/tddd17_le07.pdf [2008-04-05]

[5] Nikita Borisov, Ian Goldberg, David Wagner. Intercepting mobile communications: The Insecurity of 802.11. http://www.isaac.cs.berkeley.edu/isaac/mobicom.pdf [2008-04-02]

[6] Nancy Cam-Winget, Russ Housley, David Wagner and Jesse Walker. Security Flaws In 802.11 Data Link Protocols. http://www.ida.liu.se/~TDDD17/literature/p35-cam_winget.pdf [2008-04-03]

[7] Glenn Fleishman. WPA's Little Secret http://wifinetnews.com/archives/002453.html [2008-04-04]

[8] Aircrack-ng, Tutorial http://www.aircrack-ng.org/doku.php?id=aircrack-ng_suite-under-windows_for_dummies [2008-04-06]

[9] Wi-Fi Alliance http://www.wi-fi.org/knowledge_center/kc-8021x [2008-04-22]

[10] Wi-Fi Planet http://www.wi-fiplanet.com/tutorials/article.php/1447501 [2008-04-23]

[11] Secure management of IEEE 802.11 Wireless LANs http://softwarecommunity.intel.com/articles/eng/1090.htm [2008-04-23]

[12] 802.11b: Exploiting Flawed Security for Fun and Profit http://www.baudburn.com/files/tech/talks/802.11Hacking.pdf [2008-04-23]

[13] Techworld.com – 802.11w security won't block DoS attacks http://www.techworld.com/mobility/features/index.cfm?featureid=2599 [2008-04-23]

[14] Aircrack-ng, compatible_drivers http://www.aircrack-ng.org/doku.php?id=compatibility_drivers [2008-04-16]

[15] HCL:Wireless - Offensive-security.com http://backtrack.offensive-security.com/index.php/HCL:Wireless#Wireless_Cards_And_Drivers [2008-04-16]

[16] Remote-Exploit.org - Supplying offensive security products to the world http://www.remote-exploit.org/backtrack.html [2008-04-18]

[17] Scott Fluhrer, Itsik Mantin and Adi Shamir. Weakness in the Key Scheduling Algorithm of RC4. http://www.drizzle.com/~aboba/IEEE/rc4_ksaproc.pdf [2008-04-23]

[18] AT & T Labs Technical Report TD-4ZCPZZ. Using the Fluhrer, Mantin and Shamir Attack to Break WEP. http://cnscenter.future.co.kr/resource/hot-topic/wlan/wep_attack.pdf [2008-04-23]

[19] Technische Universität Darmstadt. Computer Science. Cryptography and Computeralgebra http://www.cdc.informatik.tu-darmstadt.de/aircrack-ptw/ [2008-04-23]

[20] Kestas, Kuliukas. How Rainbow Tables work http://kestas.kuliukas.com/RainbowTables/ [2008-04-23]

[21] (ISC)^{2}, Security Transcends Technology https://www.isc2.org/cgi-bin/content.cgi?page=738 [2008-04-23]

[22] Aircrack-ng.org Documentation http://www.aircrack-ng.org/doku.php#documentation [2008-04-24]

[23] The Shmoo Group: Airsnort Homepage http://airsnort.shmoo.com/ [2008-04-24]

[24] Wirelessdefence.org: coWPAtty Main Page http://wirelessdefence.org/Contents/coWPAttyMain.htm [2008-04-24]

[25] The Shmoo Group: Airsnart – A rogue AP setup utility http://airsnarf.shmoo.com/ [2008-04-24]

[26] Wireshark: Go deep – Frequently Asked Questions http://www.wireshark.org/faq.html [2008-04-24]

[27] ettercap http://ettercap.sourceforge.net/ [2008-04-24]