TDDC03 Projects, Spring 2007

Practical WLAN Security

David Johansson
Alexander Sandström Krantz

Supervisor: David Byers

Practical WLAN Security

David Johansson Alexander Sandström Krantz
Linköpings universitet, Sweden
davjo390@student.liu.se, alexander.sandstromkrantz@gmail.com

Abstract

In this report we have studied how 802.11 wireless networks function, with focus on the security mechanisms commonly in use today (WEP, WPA-PSK). We have done some practical tests to verify well-known vulnerabilities and our conclusion is that WEP is almost useless, and that the security WPA-PSK offers depends on its configuration. But regardless of the chosen security mechanism, 802.11 networks are always vulnerable to Denial of Service attacks, which are extremely easy to perform.

1. Introduction

The convenience of wireless networks has made them very popular and they are widely used today, for example in businesses, universities and public areas, but also in private homes. Security is an important issue in wireless networks since they use radio waves over the air as transmission medium, which means that anyone with a radio receiver/transmitter can receive and transmit data. This is a threat to the confidentiality of the information sent in these networks, and to solve this an encryption mechanism called WEP, Wired Equivalent Privacy, was developed as the original security service. As the name implies, it was intended to give wireless networks an equivalent level of security against eavesdropping to wired networks. However, WEP was later found to be vulnerable to several attacks and is now considered to be of little value as a real security mechanism [1].

This paper will examine how WEP works and how it can be broken, it will also examine WPA (Wi-Fi Protected Access), which is an interim standard that is used while the new 802.11i standard is in the process to become available. The threat of Denial of Service attacks on wireless networks will also be covered. We will start with a background section explaining how wireless networks and their security mechanisms work, which will be followed by a theoretical examination of the security vulnerabilities in these. We will then describe some practical attacks that we have done to verify the discussed vulnerabilities and the results from these. Finally we will analyze our results and give our conclusions regarding practical security in wireless networks.

2. Background

In this section we give a brief overview of how wireless networks work and the 802.11 protocols works. Focus will mainly be on the security aspects, in particular WEP and WPA.

2.1 802.11 Standard

The class of standards for wireless LANs known as IEEE 802.11 are the ones mainly used in production environments today, and will thus be the ones that we focus on in this report. The three standards in the 802.11 family used today are 802.11a, 802.11b and 802.11g. A new standard called 802.11n is on its way but is still far from production environments.

These three standards share many characteristics; they use the same medium access protocol (CSMA/CA) and the same frame structure at the link-layer. They all have the ability to reduce their transmission rate to gain extra distance and they support both infrastructure and ad hoc mode. This report will mainly focus on infrastructure mode wireless networks, because they are the most commonly used today. The three standards in the 802.11 family do have some major differences at the physical layer, but these differences are mainly about speed and the radio frequencies used. [2]

2.1.1 802.11 Architecture

The fundamental building block of a 802.11 wireless LAN in infrastructure mode is the basic service set (BSS), which contains one or more wireless stations (STAs) and a central base station known as the access point (AP). The infrastructure mode refers to the infrastructure based on the APs along with the wired Ethernet infrastructure that connects the wireless LAN to a router. In a larger network you might need more than one BSS, and you're then able to form an extended service set (ESS), which is made up of one BSS or more. The network administrator will assign a human readable Service Set Identifier (SSID, also known as ESSID) to the access point and a channel number.

In 802.11 all wireless stations need to associate with an AP before starting to send or receive frames containing network layer data. Associating with an AP means that the wireless station will create a virtual wire between itself and the AP. To be able to find the AP's, the 802.11 standard requires them to send beacon frames including their SSID and MAC address. When creating an association the AP might require the wireless station to authenticate itself and there are a number of alternatives available with 802.11. One of them is access control based on the MAC address of the wireless station. This is a commonly used, but not very secure, authentication method. [2]

2.1.2 Multiple Access Protocol

Because there might be several wireless stations associated with the same AP, trying to use the same channel at the same time, there's a need for a multiple access protocol (MAC). In 802.11 the designers chose CSMA with collision avoidance (CSMA/CA). CSMA stands for “carrier sense multiple access” and basically means that a station senses the channel for activity before trying to use it. This protocol has great similarities to the one used in Ethernet (CSMA/CD), but one big difference is that in 802.11 there is no collision detection, only avoidance. This is both because it would be expensive to build hardware able to detect collisions in a wireless network, and that you would still not be able to detect all collisions because of hidden terminals and fading. Hidden terminals and fading basically means that a wireless terminal might not see all other wireless terminals because of the nature of radio waves, so even if you have hardware which provides collision detection, it's not guaranteed to sense all activity in the network.

The risk of loosing frames in transmissions is in 802.11 dealt with using link-layer acknowledgement, if a frame passes the CRC (Cyclic Redundancy Check) test the receiver waits a period of time known as the Short Inter-frame Spacing (SIFS) and then sends back an acknowledgment frame.

To deal with hidden terminals the 802.11 MAC protocol has included a scheme that helps avoiding collisions using two new types of frames called RTS and CTS. When a wireless station wants to send a new data frame it first sends a short Request to Send (RTS) frame to reserve access to the channel. In this frame the sender indicates to the AP how much time it needs, the AP then broadcasts a Clear to Send (CTS) frame which all wireless stations associated with the AP will (probably) receive. Now no other wireless station will use the channel during the reserved time and collisions are avoided. This scheme is optional and often there's a RTS threshold value set at the wireless station, which determines how large a frame has to be to use the scheme. [2]

2.1.3 802.11 Frames

The frames used by 802.11 are very similar to those used in Ethernet, but has a number of specific fields. As with Ethernet there's a payload, typically consisting of a IP datagram, and a cyclic redundancy check (CRC) value used to detect bit errors.

One of the major differences between the Ethernet link layer frame and the one used in 802.11 are the address fields, which in 802.11 frames are four fields instead of two. The fourth field is used for networks in ad hoc mode and the third field is needed for internetworking purposes, especially connecting the access point to a router.

In infrastructure mode the first address field is the MAC address of the receiving wireless station, the second field contains the MAC address of the sending station and the third field the MAC address of the router connecting to the access point. Because the access point is a link layer device it doesn't understand IP addresses, so when moving a datagram from a wireless station to a router it needs to know the MAC address of both the sender and the router. It's still not enough to have only this information in the link layer frame because we also need to know the MAC address of the access point in the link layer protocol used in 802.11.

Except for the data frames and control frames (RTS, CTS, ACK) mentioned before there are also management frames used to establish and maintain connections. These include, among others, frames to authenticate, deauthenticate, associate, reassociate and deassociate. In this report we're particularly interested in the deauthentication and deassociation frames as these can be used to launch a DoS attack without being authenticated to the network. This is possible because these frames are sent and received unauthorized in all 802.11 protocols used today. [2]

2.2 WEP

WEP, *Wired Equivalent Privacy*, was the first security mechanism that came with the 802.11 standard for providing data confidentiality equivalent to wired networks. It is optional to use in wireless networks, and the intention was to protect from casual eavesdropping [3]. This is achieved by encrypting the data using a symmetric stream cipher called RC4 from RSA Data Security, Inc. The standard doesn’t specify how the shared, secret key should be transmitted to the STAs, but assumes that they have been delivered via a secure channel independent of the 802.11 protocol. The original standard specified that a 40-bit[1] key should be used, but many vendors today support 104-bit[2] keys to improve resistance against brute-force attacks. The reason for using only a 40-bit key in the original standard may be due to the fact that the designers tried to design WEP so that it could be approved for export outside of the USA.

Encryption with WEP is done by doing a bit-wise XOR of the bits in the data stream with the bits of a pseudorandom key stream. This key stream is generated by the RC4 algorithm, which uses a 24-bit IV (Initialization Vector) and the secret key as its seed. The purpose of the IV is to extend the useful lifetime of the secret key by creating a new seed for every new IV while using the same key. The IV is sent as clear text in the beginning of the 802.11 frame body so that the receiver who knows the secret key can always reconstruct the key stream and thereby decrypt the message. The IV is changed periodically, and it is recommended that it be changed for every new MPDU (MAC Protocol Data Unit) so that a known sequence of the key stream from one MPDU doesn't give the content of another [3].

Encryption is done over the data field and a 32-bit Integrity Check Value (ICV). The ICV is a CRC-32 (Cyclic Redundancy Code), which purpose is to detect bit errors in transmission. By encrypting it together with the data, it was expected to also serve as an integrity protection mechanism.

Decryption of a WEP-encrypted frame body is done by first extracting the 24-bit IV and concatenate this with the secret key to form the seed to the PRNG. The resulting key sequence is then XORed with the ciphertext and this will give the plaintext data and ICV. A new ICV is calculated on the decrypted plaintext and compared with the ICV contained in the message, to verify that the decryption process was successful. If they don't match an error indication is sent and the packet is not passed on to the LLC layer [3].

Since WEP is a symmetric cipher, both encryption and decryption is done with the same secret key, but it is not necessary to use the same key for all frames. A key ID is sent with the IV and it is used to select one of four possible secret keys for decrypting the frame body. Another possibility is to have different WEP keys for each pair of STAs, by using an array that maps each key to a specific MAC address [3].

WEP encryption is also used for Shared Key Authentication in 802.11 networks. The STA that wants to connect to the network first sends a request to the AP, which returns a random 128-byte authentication challenge text. The STA then has to encrypt this challenge using WEP and send it back. The AP decrypts the message and compares it with the challenge text that was sent before, and a final frame is sent telling if the authentication was successful or not [3]. In this way, only those STAs that know the shared secret key will be granted access to the network.

2.3 WPA

When the world realized that there were serious security vulnerabilities in the WEP implementation, which we describe in section 3.1, both users and developers demanded something better. Because of this, work on a new standard called 802.11i (also known as WPA2) started, but some felt that this work was too slow and at the same time there was a demand to be able to easily upgrade existing hardware to be more secure. 802.11i was going to use AES instead of RC4 for encryption, so the old hardware would be unable to support 802.11i. Because of this the Wi-Fi Alliance created a new security system called Wireless Protected Access (WPA) based on parts of 802.11i, but with the ability to use existing WEP hardware [4].

To create a strong security mechanism based on WEP-enabled hardware, a new protocol called Temporal Key Integrity Protocol (TKIP) with three new elements was included in WPA. These elements were a new message integrity code (MIC) called Michael, a new packet sequence discipline and a per-packet key mixing function. WPA uses RC4 with a 128 bit base key and 64 bit authentication key together with a 48-bit IV and the already mentioned key mixing function. Michael protects source and destination addresses and enforces IV sequencing and the IEEE 802.1X protocol manages the keys.

An algorithm calculates a MIC at the transmitter using a function with the key and data as input and sends its output together with the data to the receiver. At the receiver the same calculation is made and the result is compared with the MIC included in the received data. Conventional MICs (such as HMAC-SHA1 used by IPSec etc) can't be used on existing hardware as they are too heavy, because of this a new MIC called Michael was invented for WPA. With this MIC algorithm there are some performance degradations but there are no known algorithms with less computational power needed that provides appropriate security. Michael uses shifts, XORs and additions to create the final 64-bit authentication tag sent with the data. The security level of a MIC is measured in bits, and Michael has a security level of only 20 bits, which makes it too insecure on its own. To solve this problem WPA requires new keys if a MIC fails to validate, this rekeying is limited to once per minute and

[1] This 40-bit key together with the 24-bit IV forms the key used in what's known as 64-bit WEP Encryption.

[2] This key together with the IV is used in what's known as 128-bit WEP.

with this configuration the number of false positives are expected to about one per year.

To prevent replay attacks WPA extends WEP with a 48-bit sequence number, which is mixed with the encryption key because of implementation constraints. This design creates ICV (Integrity Check Value) or MIC failures upon replay attacks. To prevent the FMS (Fluhrer-Mantin-Shamir) attack on WEP there is a per-packet key mixing function introduced in WPA, this function uses the base key, transmitter MAC address and the packet sequence number as input to create a per-packet key.

802.1X is used to provide WPA with fresh keys for encryption (128-bit base key) and authentication (64-bit key used by Michael) for each new association [5].

3. Security issues in wireless networks

This part of the report discuss the various security issues that are present in wireless networks, and gives a theoretical background to weaknesses that can be exploited in practice.

3.1 WEP Vulnerabilities

Today WEP is considered almost useless as a security mechanism in wireless networks and is, or at least should be, replaced by other security systems. The flaws in the WEP design are the following [1]:

- The IV's are too short (only 24 bits).
- The CRC checksum (ICV, Integrity Check Value) is insecure.
- The combination of the IV with the key is done in a way that enables crypto analytic attacks.
- There is no integrity protection of the source and destination addresses.

One of the first known attacks to work on WEP was the brute force attack, where you simply try all possible keys. The initial standard specifies a 40-bit key and this is short enough to enable a brute force attack to success in weeks on a modern machine. There are also specific implementations, which decreases the key space even more by using methods to translate human readable input to WEP keys in bad ways. To solve the brute force problem most of the vendors have now increased the key size to 104 bits.

Integrity protection is supposed to protect a possibly known message from tampering, but this fails for WEP. An attacker can get the key stream by XORing the known plaintext and its ICV with the cipher text, then create a new text, calculate its ICV and simply XOR that with the key stream to form a perfectly valid WEP-encrypted message. All without having to know the secret key. It is a common knowledge in cryptology that stream ciphers should not be used for achieving data integrity [6].

It has been proved through cryptanalysis that security of WEP is independent of the key size, which means that the mentioned key size incrementation didn't actually increase security even though it made brute force attacks infeasible. If a key stream is recovered it's possible to both decrypt data, which was encrypted using this key stream, and to send new data. When first discovered, it was considered hard to know the plain-text of a packet so this vulnerability was regarded unimportant. Later on ways to discover a key stream was found, these ways use the fact that the 802.11 frames contains well known plaintext. Together with fragmentation of packets it enables an attacker to insert encrypted packets into a network, without knowing the real key, with only a very short known key stream. This could, for instance, be recovered from the first 8 bytes in the package header, which are known.

It has also been discovered by Fluhrer, Mantin and Shamir that because of weak IV's in some packets it’s possible, by gathering ~ 1.000.000 packets, to calculate the key. The first automated tools to compromise a WEP network used this vulnerability. The vendors patched this problem by disabling these weak IV's in their hardware, but it turned out that there were more weak IV's than the once discovered at first and new filters had to be implemented. Of course not all hardware have all these filters included, so the problem is still there. To gather enough packets could take days and therefore this attack was from the beginning hard to use.

The problems with reliably recovering the key stream and to speed up the IV attack have both been solved today. There are methods to recover one byte of the key stream by sending at most 256 packets, for instance by using the fact that the AP will forward the packet if you guess the next correct byte in a key stream which is partly known (which all are, as mentioned before) [1].

To speed up the IV attack there are several ways, one is to replay packets to create more traffic and decrease the time it takes to gather enough packets. To solve this some vendors implemented key-changing schemes so that the attacker wouldn't have enough time. But of course also this obstacle has been “solved”, today it's possible to attack the network fast enough. There are attacks available today that breaks a 104-bit WEP key with 95% probability using only 85.000 packets. With packet replay this is done on a modern machine in less than a minute [7].

3.2 WPA Vulnerabilities

Even though WPA corrected the problems with WEP, new vulnerabilities have been discovered. Some even say that WPA (with pre-shared keys) is easier to crack than

WEP, and with pre-compiled dictionaries it's also very fast [8]. Another vulnerability is related to the MIC used by WPA (Michael), as a countermeasure to attacks it turns off the access point for one minute if it detects two failed forgeries [9]. According to Niels Ferguson, the inventor of Michael, this angle of attack was well known during the development, but there are other easier ways to launch a DOS attack on a wireless network based on 802.11, so therefore this was not considered a major threat [10].

In the beginning of a transmission using WPA-PSK there's a four-way handshake to set up the encryption, the information sent within this handshake can be used in an offline dictionary attack. In this handshake a transient key is used to produce a hash of the frames, because of this any program available for offline dictionary attacks against hashes (and there are lots of them) can be altered to use information from the handshake to perform an attack. The WPA standard states that the users should choose passwords longer than 20 characters, but most people are unwilling to do this for practical reasons. Shorter passwords are unable to deter an offline dictionary attack [11]. With a program to crack the password, like coWPAtty[3], it's possible to use pre-compiled hash tables in the dictionary attack. This increases the speed of the attack and makes it a more usable attack. There are even pre-computed tables available for download for the most common SSIDs.

3.3 Denial of Service Attacks

Most of the literature on security in wireless networks has been focused on vulnerabilities in the confidentiality, integrity and authentication mechanisms. But another, often equally important, threat comes from Denial of Service (DoS) attacks, making the users are unable to transmit and/or receive packets over the wireless link. Bellardo and Savage [12] divide the DoS vulnerabilities of 802.11 networks in their report into two categories: *Identity* and *Media Access* vulnerabilities.

3.3.1 Identity Vulnerabilities

Identity vulnerabilities are possible because of the implicit trust in a sender's reported source address and the fact that management frames are not authenticated. This enables an attacker to spoof messages that will result in a DoS attack. One such attack is the deauthentication attack, in which an attacker spoofs a deauthentication request, pretending to come from either the victim's STA or the AP. This will stop the data transmission between them until they have successfully authenticated again. Repeating this deauthentication message can effectively shut down the service. A single individual station as well as the whole wireless network can be the target of this attack.

Another attack, that is very similar to the deauthentication attack, is to spoof disassociation messages, which in essence will have the same effect in denying a victim access to the wireless network. This is, however, a little less effective since the victim station can get back on to the network faster after being disassociated than after being deauthenticated. The attacker therefore needs to spoof more messages to keep the victim off the network for the same amount of time compared to a deauthentication attack.

802.11 also provides a power-saving feature by which a station can announce that it's going to sleep-state, and the access point will then buffer all messages to this client until it wakes up again and polls the messages. The data is then erased from the buffer in the AP. The AP and the STA are synchronized so that the STA knows when to wake up and check for the signal that announces that new messages are waiting. If there are no messages, it goes back to sleep again. This offers many ways to launch a DoS attack, such as polling a victims messages off the buffer before it awakens, spoof fake packets telling there are no new messages or disrupting the time synchronization so that the victim STA wakes up at the wrong time.

3.3.2 Media Access Vulnerabilities

A different class of DoS attacks on wireless networks has to do with the mechanisms it uses to avoid collisions on the physical medium. One of these mechanisms is the *Short Interframe Space* (SIFS), which is a short period after each sent frame that a station must wait before it can send the next frame. The sending station has to begin a new wait cycle if any traffic is sensed on the network during this time, so an attacker could launch a DoS attack by continuously sending a small signal before the end of each SIFS. This would make the channel unavailable to all other stations, but the drawback with this method is that it requires a lot of packets to be sent since the SIFS period is so short. On a 802.11b network it is only 20 microseconds which would require 50,000 packets per second to be sent [12].

A much more effective attack is to use the *Duration* field of a 802.11 frame. The purpose of this field is to set the time needed for transmitting this frame so that other stations won't try to send while it's still transmitting. All stations that receive a frame with a value set in the duration field will update their *Network Allocation Vector* (NAV) to this number, which is then decremented as time elapses. A station may only send when its NAV has reached zero, so an attacker can take down the

[3] http://www.wirelessdefence.org/Contents/coWPAttyMain.htm

network by repeatedly sending a frame with a large value in the duration field. This attack would only require about 30 packets a second to take out a channel completely on a 802.11b network since the maximum NAV time is about 32 milliseconds. This technique is especially useful with RTS-frames, since a station which receives it must answer with a CTS-frame containing the duration field, which will expand the range of the attack not only to those within the reach of the attacker but also to all stations within the reach of the target station receiving the RTS-frame [12].

3.3.3 Radio Interference

A little more “brutal” attack for disrupting the service of a wireless network is to jam the radio signal by using a transmitter that overpowers the signal between the AP and the STAs. 802.11b and 802.11g networks use the 2.4 GHz band for transmitting its signals, which is the same frequency band that Bluetooth, cordless phones and other devices use. It can be enough with a pair of these devices nearby a wireless network to degrade its service. A more powerful jamming device that really could knock down a 802.11 network is simply a normal microwave oven. All that’s needed is to remove the metal shielding and extract the unit producing the microwave signals along with its power supply, and you find yourself with a 1000W radio transmitter! [13] Note that exposure to microwaves of these powers is a serious health risk and removing the shielding from a microwave oven is strongly discouraged.

4. Practical attacks on Wireless Networks

To verify some of the theoretical vulnerabilities of wireless networks that we have described, a series of practical attacks using different techniques were done, which are described in this section.

4.1 WEP Attack

In our attack on WEP we used the knowledge presented in the report “Breaking 104-bit WEP in less than 60 seconds” [7], and the tool *aircrack-ptw* they created to crack the WEP key. We set up a wireless network with one access point and a client station associated with this using the correct WEP key. We also used an attacking station, which knew nothing about the network from the beginning.

To make the situation as close to real life as possible we first set up the wireless network and the client station authenticated with the access point before we initiated our attack. The first thing we did on the attacking station was starting up *Kismet*, a program to monitor and capture wireless traffic, which found the network, as well as the clients associated with it, in a second or two. The client needs to send some kind of data for Kismet to be able to find it, real life wireless networks in use will probably have clients sending data almost all the time so this won't be a big issue.

Now that we knew the client and the access point’s MAC addresses we used the program *aireplay-ng*, which is able to inject packets into a wireless network, to replay ARP packets. To be able to send ARP packets into the wireless network without knowing its WEP key you need to capture at least one ARP packet sent from a station associated with the access point. When you've got this ARP packet you can inject it back into the network as it is, the access point will now broadcast this ARP packet to all stations using a new IV. You might consider the fact that you need to capture at least one ARP packet as a problem, but again, in real life wireless networks there are probably clients trying to access new IP addresses, and thus creating ARP packets, on a regular basis. Now that aireplay-ng has got an ARP packet it's able to flood the network with ARP traffic, and with help from the access point all this ARP packets will get new IV's. We managed to insert about 300-400 packets/second, if lucky you should be able to insert at least 1000 packets/second but this depends on the hardware and software implementation.

When we thought that we had gathered enough packets we passed our Kismet dump file on to the program aircrack-ptw. According to the developers this program will recover a 104-bit WEP key in 95% of the cases with only 85,000 packets. It uses a method created by Klein [14] to attack the RC4 algorithm without the requirement of weak IV's, but specialized to WEP by Tews et al.

In our first set up we used a 40-bit WEP key and with the described method we were able to recover this key with 20,000 gathered packets. As mentioned before we injected about 300-400 packets per second, so gathering these packets took us about 1 minute. After this we decided to increase the key size to 104-bit, still using WEP and with the same method we now needed 50,000 packets to recover the key. With the same injection speed this took us about 2,5 minutes.

Our results were better than we had hoped for. Breaking a WEP network in minutes without any requirements other than that there should be at least one station associated with the access point which sends at least one ARP packet while we're monitoring the network was almost unbelievable. The results also show that the number of packets needed to recover a key doesn't scale with the key size, so choosing a larger key size is not a solution to the problem.

4.2 WPA Password attack

A similar setup to the one used during our WEP attack was used for this experiment, but the Access Point and client station were set up to use WPA with a pre-shared key instead of WEP. The SSID for the Access Point was set to *default,* which is the third most common SSID according to wigle.net [15]. The first step by the attacker was to send a deauthentication frame using the program *aireplay-ng*, which made the client station to loose network connection and forced a reauthentication. During this time we used Kismet to capture the 4-way authentication handshake. The program *coWPAtty* was used together with a pre-computed table of hashes for our SSID and a dictionary containing around 172,000 words. The reason for using a pre-computed table of hashes is to speed up the test of passwords to around 18,000[4] keys per second, which is a huge improvement compared to the modest 12 keys per second without the use of tables [16]. This attack is highly dependant on the quality of the dictionary used with the pre-computation of the hashes, and our first two tests with the passwords set to *myownwireless* and *secretadmin* failed to recover the key. This was somewhat surprising since both passwords consisted of common words, not unlikely to be in a password. The dictionary has some clear limitations when it comes to finding more complex passwords not only consisting of a single word. But the attack tool was very fast at recovering the password when we chose a password that we knew was in the list, it was a matter of few seconds. This means that networks using WPA-PSK with weak passwords and common SSIDs can be broken very fast, and only requires one captured authentication process, which is easy to get hold of.

4.3 Denial of Service Attack

The Denial of Service attack that we tried in practice was the *Deauthentication attack* described in 3.3.1. The tool we used for this was *aireplay-ng,* which includes the option for sending deauthentication frames. The attack was launched from a laptop running Linux with a Prism-based wireless network card, and the target was a laptop running Windows XP. The target machine was continuously sending PING-packets to the Access Point so that we could easily verify the connection. This attack was highly effective and the targeted station lost the connection as soon as the deauthentication frame was sent. After a single deauthentication, it took roughly 5-10 seconds before it was authenticated again and could use the network. Sending multiple deauthentication frames with a short interval between them kept the target station off the network for as long as the attack was going on. However, when the attack stopped it managed to recover network connection again after a little while, so it was at least not causing a system crash, which was reported for a HP Jornada Pocket PC in [12].

5. Conclusions

It has been known for many years now that WEP has some vulnerabilities that makes it insecure, and the algorithms for breaking it has become more effective lately. The first algorithms needed a few million packets to break WEP, later on improved algorithms required only about 500,000 packets. Today, the most advanced algorithm can break WEP with less than 100,000 packets, which is easily captured within minutes with an active attack using packet injections. Our tests shows that it is not only fast but also easy to break both 40-bit and 104-bit WEP with tools existing today, so the only thing WEP protects against is from users to unintentionally join the network. Anyone with motivation and a little knowledge will succeed to break a WEP encrypted wireless network.

WPA with pre-shared keys has a vulnerability that can be used for a dictionary attack on the password. The strength of WPA is therefore dependent the entropy and length of the password. Using a non-standard SSID will stop the attacker from using downloaded pre-computed tables of hashes, but it will not stop the success of the attack itself, it is only a matter of how long it will take. If the WPA password is in the dictionary that is used for the attack, it will be recovered. As the attacks on WEP have become a lot more effective during the years we also believe that this could be the case for WPA-PSK in the future.

Denial of Service attacks on 802.11 networks are so effective and easy to perform that we cannot do anything but agree with Niels Ferguson in his statement “Anyone using a wireless network for mission critical use is out of his mind, and I consider it to be criminally negligent.” [10]

Our recommendation to users of small wireless networks is to use WPA with a pre-shared key of random characters with a length of at least 20. Since the attacker is unlikely to have physical access to the Access Point it is better to have a strong password that is written down on a note under the AP than to have a weak password that is easy to remember. Since all the security mechanisms are meaningless if an attacker gains access to the configuration panel of the wireless access point, it’s important to protect it with a strong password, and preferably only allow local access to it. During our tests we found many wireless routers and access points with no password or a standard password, which allowed complete control over them.

[4] This was on a P3/700 laptop and the number would probably increase several times on a modern computer.

References

[1] Andrea Bittau, Mark Handley and Joshua Lackey, *The Final Nail in WEP's Coffin*, 2006.

[2] James Kurose and Keith Ross, *Computer networking – a top down approach featuring the Internet*

[3] IEEE 802.11 Standard, 1999 (Reaffirmed 2003)

[4] WPA 4-29 White Paper, Wi-Fi Alliance, April 29 2003

[5] N. Cam-Winget, R. Housley, D. Wagner, and J. Walker, *Security Flaws in 802.11 Data Link Protocols*

[6] Fåk, Viiveke, *Course Notes in Cryptology*, p. 21, Institutionen för systemteknik. Linköping, 2004.

[7] Erik Tews, Ralf-Philipp Weinmann, Andrei Pyshkin, *Breaking 104-bit WEP in less than 60 seconds*, Technische Universität Darmstadt, 2007.

[8] Seth Fogie, *Cracking Wi-Fi Protected Access (WPA)*, Part 1, March 4 2005.
URL: http://www.informit.com/articles/printer friendly.asp?p=369221&rl=1 Accessed 2007-04-25

[9] Elisa Batista, *Wi-Fi Encryption Fix Not Perfect*, URL: http://www.wired.com/techbiz/media/news/2002/11/56350 Accessed 2007-04-25

[10] Ferguson, Niels, *Re: DOS attack on WPA 802.11?*, The Cryptography Mailing List, Nov 2002. URL: http://www.mail–archive.com/cryptography@wasabisystems.com/msg03078.html Accessed 2007-04-26

[11] Glenn Fleishman, *Weakness in Passphrase Choice in WPA Interface*, November 4, 2003.

[12] Bellardo, John, Savage, Stefan, *802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions*, Department of Computer Science and Engineering, University of California at San Diego, 2003.

[13] Sawicki, Ed, *Wireless and Denial of Service Attacks*, 2002 URL: http://www.biznix.org/articles/wirelessdos.html Accessed 2007-04-25

[14] Andreas Klein, *Attacks on the RC4 stream cipher*, submitted to Designs, Codes and Cryptography, 2007.

[15] http://www.wigle.net/gps/gps/Stat Accessed 2007-04-26

[16] RenderMan, *Church of Wifi WPA-PSK Rainbow Tables*, RenderLab
URL: http://www.renderlab.net/projects/WPA-tables/ Accessed 2007-04-26