Practical WLAN security

Eldeklint, Jon Gunnbäck, Johannes
Email: {jonel563,johgu901}@student.liu.se
Supervisor: David Byers, {davby@ida.liu.se}
Project Report for Information Security Course
Linköpings universitetet, Sweden

Abstract

This report covers the basics security standards in today's wireless networks, how they work, existing flaws and vulnerability's. We will go trough the basic architecture of 802.11, the security it provides like WEP, WPA and WPA2. Further more we will go trough some practical experiment exploiting the flaws described in the article. The result of our practical experiments show upon how easy it is to break encryptions and deploying Evil twins, even with very little understanding of the security structure.

1. Introduction

Where ever you go, either it's an workplace, coffeeshop, library or even a park there is a high chance today that you'r able to connect to wireless networks. However, with the rising accessibility of Wi-Fi, this also makes attacks more likely to occur, both from intentional and non-intentional attackers. Intentional as hacking into your network or non-intentional when you connect to the wrong accesspoint. Our goal with this paper is to show how easy it is to exploit vulnerabilities in the wireless networks of today.

We'll describe the fundamental architecture of 802.11 networks and the security it provides. To give a higher understanding for some of the problems that exists with the security standards. With easily accessible software and tools we'll show two experiments that exploits these vulnerabilities. First in line is how weak the old security standard WEP is. The second experiment will be to deploy an Evil Twin accesspoint to show that there is not only the technology that is unreliable.

2. Background

In this chapter we'll cover basic architecture in 802.11 networks and the security standards it provides.

2.1 802.11 Standards

The architecture of the wireless LANs is specified by the 802.11 standard created by IEEE [1]. There are a few versions of the standard with differences in frequency and speed. Briefly these are a few of the current standards

802.11 (1997): 2.4Ghz, 2Mbps

802.11a (1999): 5Ghz, 54Mbps

802.11b (1999): 2.4Ghz, 11Mbps

802.11g (2003): 2.4Ghz, 54Mbps

Even when things like speed and frequency differ, most things are the same as the original 802.11 standard. They all are using the same medium access protocol, CSMA/CS and have features for increasing range by the cost of speed. They also support the two connection modes of ad-hoc and infrastructure, but since this reports is about practical WLAN security we'll not go any further in on that and just care about the infrastructure mode.

2.1.1 Architecture

In a wireless LAN in infrastructure mode the mainstay is basic service set. The BSS is containing the wireless stations that can be anything from laptops to mobile phones. Those stations are connected to an accesspoint.

2.1.2 Frames

The 802.11 standard define a lot of different frame types that wireless stations uses for communication, managing and controlling the link. All frames has apart from fields about sender and destination station a control field that contains information about 802.11 protocol version and other things like if some encryption is turned on, etc. There are also fields for frame sequence numbers and error checksum.

Apart from the normal data frame 802.11 specifies three common control frames, the Request to Send (RTS), the Clear to Send (CTS) and the Acknowledgment (ACK) frame. The RTS/CTS starts the transmission by requesting for channel time and receives permission to send from the target with a time slot that makes all other stations to hold off transmission for that time. More interesting than the control frames are some of the management frames:

Authentication: Authentication in 802.11 is for identifying a station to the access point and se if it's accepted to connect. It also serves for making a secure connect over WEP or so via a challenge-respond sequence.

Deauthentication: A station sends a deauth frame to another station if it wants to terminate the secure session.

Association request: enables a accesspoint to allocate resources for a new station. The frame contains information about the station and what SSID (Service set identifier) it wishes to connect to.

Association response: the accesspoints response to an association request. If the accesspoint accepts the connection the frame contains information about the association like supported data rates and it's association id.

Reassociation request: if an station is on the move and finds an other accesspoint with a better beacon signal, the station will send a frame for reassociate with the new accesspoint. The new accesspoint is suppose to handle forwarding of eventual data frames buffered by the old accesspoint.

Reassociation response: like the normal association responde frame this frame contains information if the connection is accepted. More information about the association like it's id etc. is sent.

Disassociation: A station sends one of those frames to another station if it wants to terminate the session.

Beacon: The accesspoints send in intervals information in beacon frames about that it exists and relay information like SSID and timestamp

Probe request: A station sends one probe request when it wants to know more about another station, for example a client might send a probe to find accesspoints in it's range.

Probe response: As an answer to the request a station can send a probe response containing information about capability, supported data rates and more. [2]

2.2 WEP (Wired Equivalent Privacy)

The Wired Equivalent Privacy protocol was introduced into the 802.11 network standard to provide the same level of security as in a wired network. To be able to achieve this there are three main goals with WEP that needs to be enforced:

Confidentiality which is intended to prevent a possible attacker from eavesdropping. Encryption is applied to achieve this.

Access control to protect access to the wireless network from the wrong users. A feature included in the 802.11 standard is to drop all packets not correctly encrypted with WEP.

Data integrity to prevent tampering with transmitted messages. WEP uses a integrity checksum for this.

To fulfill the above goals, as mention before, WEP is using checksums and encryption. WEP relies on a shared secret key k shared between all parties in the communication. Here follows a short description of how the encryption algorithm works:

First a integrity checksum is calculated c(M) on the message M. The message and the checksum is put together to form the plaintext P=<M,c(M)>. After the plaintext is created it will be encrypted using RC4. A initialization vector (IV) v is chosen and together with the shared key k the RC4 algorithm will generate a long sequence of pseudorandom bytes i.e. the keystream RC4(v,k). When both the plaintext and the keystream has been created a exclusive-or is preformed to preduce the ciphertext C. This is denoted C=XOR(P,RC4(v,k)). Finally the IV and ciphertext is transmitted from sender A to receiver B, A→B : <v,C>. When the packet arrives at the receiver it will be decrypted simply by reversing the encryption process. First regenerate the keystream RC4(v,k) and XOR it against the cipher text P=XOR(C,RC4(v,k)). Finally the receiver will split P into <M,c> and recalculate the checksum c(M) and compare it the checksum in the message to validate that its the right message received. [3].

2.3 WPA (Wi-Fi Protection Access)

It was quite fast obvious that WEP had some major problems so IEEE started to work on a new security standard named 802.11i but it had taken far to long time to wait for IEEE to complete the new standard before securing the wireless networks. So instead of waiting for a new standard that would require new hardware because of the switch of encryption algorithm, a fix that combined parts of the new standard with the old hardware had to be made. In 2002 the Wi-Fi Alliance combined the TKIP (Temporal key integrity protocol) of 802.11i with the RC4 cipher of WEP. To protect WPA against the weaknesses in WEP a set of algorithms are used in TKIP like the Message Integrity Code for avoiding forged packages, but since the abbreviation MIC already is used, the algorithm is called Michael instead. Michael uses a 64bit key and partitions packets into 32bit blocks, then shifting, applying XOR and additions to calculate a 64bit authentication tag.

For protection against replay attacks there is a new discipline on packet sequences, the TKIP simply mixes the sequence number into the encryption key which make a replayed packet get catched as an ICV (Integrity Check Value) or MIC failure.

For avoiding the usual cryptanalysis attacks that can be made on WEP like FMS, chopchop etc. there is a function for mixing the 128bit WEP key per packet, that takes the base key, transmitter MAC and the sequence number of the packet.

The MIC countermeasures in TKIP consists of requiring a rekey after detecting a invalid MIC and limits rekeying to one per minute this since the Michael algorithm is too weak to stand alone. However false positives is calculated to only appear about once per year. [19] [4]

2.4 WPA2/RSN (Robust Secure Network)

The latest and currently most secure feature for wireless network security today is WPA2. As in WPA the WPA2 protocol also supports IEEE 802.1X/EAP authentication or PSK (pre-shared keys) technology. The strongest difference between WPA2 and WPA is that WPA2 use AES-based algorithm CCMP(Counter Mode with Cipher Block Chaining Message Authentication Code Protocol) instead of RC4 which provides longer keys and is overall a lot stronger encryption algorithm. The drawback is that WPA2 is not compatible with current hardware for WEP and WPA and need upgrades to work. WPA2 works in two modes. Enterprise mode which is designed for larger companies and enterprises. It uses the IEEE 802.1x authentication framework and an authentication server to provide access to the WLAN. The second mode uses pre-shared keys and is designed for homes or small offices that don't have authentication servers available. Both modes work with the AES encryption algorithm.[5]

3. Vulnerabilities in wireless networks

Short cover of security flaws in wireless networks.

3.1 802.11

There is a few problems with the 802.11 standard by default one of the most annoying and difficult to tackle is the fact that since 2.4 Ghz is a open frequency a lot of devices operates there. You can find everything from DECT phones to microwave ovens on the 2.4 Ghz spectrum. Having those things around you wireless network may jam the network, at least make it's availability suffer.

The 802.11 standard use the MAC address of it's devices a lot for identification, these are however very easily spoofed.

If that wasn't enough there is an other problem, 802.11 management frames are not protected even if it's data frames are, this results in that it's possible to forge management packets which can make some trouble if the accesspoint gets flooded with association frames or stations receives false deassociation frames. This is used to make several other attacks possible.

3.2 WEP

In chapter 2.2 we described the goals for WEP and how it was provided. This means that WEP is secure right? Wrong!

Here follows a short description of the fundamental flaws in WEP and show how all three main goals is broken.

If we look pass the general weaknesses of the infrastructure in 802.11 networks that can affect WEP we have a few weaknesses in the protocol itself. The most serious problem is the RC4 algorithm and the use of so called weak keys. The RC4 algorithm is implemented in a non-standard way and uses a 24-bit public IV together with the secret key, and the IV is sent in the clear. This is enough data to perform cryptanalysis. Gain access to the secret key is all you need to be able to break all three goals in WEP. Even if the attacker don't manage to recover the key its still possible to recover all different types of keystreams. There exists 2^24 distinct keystreams, a message frame is up to 1500 bytes long which means that it only takes about 24GB of storage for all possible keystreams. One way of getting hold of keystreams is for the attacker to send packets where they know parts of the plaintext in the response.[6]

Another flaw in WEP that is easy exploitable is keystream reuse. RC4 is a stream chiper and the same key should never be used twice and this is enforced by the changing IV:s. Weak keys is a key which when used with a specific cipher, makes the cipher behave in some undesirable way, in the case of RC4 weak keys is reuse of the same key. Because the IV is restricted to 24 bits there is almost guaranteed that the same IV will be reused multiple times. For example an access point sending 1500 byte packets and achieving an average 5Mbps bandwidth will have used up all distinctive IV:s in less then half a day. If an attacker can get hold of two messages encrypted with the same IV C1=XOR(P1,RC4(v,k) and C2=XOR(P2,RC4(v,k) he can xor the two chipertexts to get XOR(C1,C2) = XOR(P1,P2). There are known techniques to get P1 and P2 given XOR(P1,P2).

Furthermore WEP is using the CRC checksum function to verify integrity. The idea with the checksum is to prevent any tampering with the message in transit. The CRC is preformed on the message and not on the ciphertext and the function itself is linear, this makes it possible to perform changes in the ciphertext without changing the checksum.[7] There are a lot of possible ways of attacking WEP but the fundamental flaws that makes the three main goals of WEP broken is the weak keys and the linear checksum function. Further reading is advised and recommended papers are the famous Fluhrer, Mantin ans Shamir attack[8] and a more up to date and improved attack against WEP[9]

3.3 WPA

Sure, WPA corrects alot of the problems with WEP but also provides some new vulnerabilities, a WPA protected network with a bad passphrase and a standard SSID will probably be even faster broken into than a WEP protected one. This since it's possible to capture the WPA 4-way handshake easily thanks to unprotected management frames. The sharing of the key is then attacked by dictionary attack. Since the WPA key hash "PBKDF2(passphrase, ssid, 4096, 256)" is quite slow to calculate since it iterates a SHA1 algorithm 4096 times [10], pre-calculated rainbow tables are instead used for speeding things up into insanity. It's not strange to be able to test about 20k hashes a second with rainbow tables [11]. Another weakness in WPA is that the Michael MIC algorithm as an countermeasure for forgeries it throws everybody out and shuts the AP down if it finds two forged packets within a minute [12] this can be used to DoS the wireless network.

3.4 WPA2/RSN

As mention before WPA2 is the strongest security feature for wireless networks to day. Does that mean that it’s unbreakable? The answer is unfortunately no. The weakness here lies with user’s tendency to use weak passwords that are easy to guess. There exist off the shelf tools that can generate brute force and dictionary attacks against WPA2. Further more the WPA2 protocol does not provide any protection against different DoS attacks such as radio frequency jamming, de-authentication, de-association etc.[13]

3.5 Evil Twins

The so called "Evil Twin" attack towards a wireless network is launched by installing another false accesspoint. The false accesspoint will have the same SSID as the victim and is supposed to have superior signal strength so the victim will connect with the false AP instead of the real one. The attacker can make this easier for her by disassociate clients instead of just waiting for new ones to connect. The false network should be designed so the victim client don't realize that she is at the wrong place, the evil twin could be connected to internet via 3G or even connected to the real network. By doing this an attacker can do everything any other MITM-attack would give room for, like listen for passwords, credit card numbers, change requested information, etc. all without the victim notices anything. [14][15]

3.6 More attacks on wireless networks

Apart from the normal key cracking attacks on WEP and WPA protected networks it's also possible to do inject packages in net encrypted by both, injecting in a WEP network is pretty straight forward, get the key send in the packages. In the WPA case it's a lot trickier, but in theory it consists of obtaining the clear text of an small packet like an ARP, this can be done with some tool for cracking WEP packets like a chopchop attack. This is possible since the payload of an arp is pretty much known. When the clear text is know, the Michael algorithm is reversed to acquire the MIC key, then since of QoS techniques it's possible to inject forged packages with the MIC key 7-15 times in a short period of time depending of how quick the network rekeys. This is however a one-way attack but ARP poisoning or make things call home should be possible. There is a tool called tkiptun-ng that does this attack but it only supports a few wireless cards and is unstable. [18][20]

Another possible attack is that the management packets in 802.11 is unencrypted and very easy to forge, therefore it's theoretical very easy to mess with a wireless network via the management packets. The way of doing so is pretty much; open a raw socket and send your forged packages with a false transmitter MAC address to some poor receiver. It's as simple as, DoS the users? Send Disassociation packets. Give the AP a lot of work? Send Authentication/Association packets in large amounts. [2]

4. Attacking wireless in practice

In this chapter we’ll describe how we were able to perform attacks on wireless networks.

4.1 Preparation and tools

Performing the practical experiments required some hardware, a couple of laptops, a wireless accesspoint and a wireless NIC that supported packet injection. The hardware is neither rare nor expensive were the wireless NIC was hardest to acquire. Hard, as in 5minutes at ebay and 10£ and replacing the internal NIC of the laptop with the new one. The choice of accesspoint fell on the Linksys WRT54g not cause we really needed all things it's capable of, but it was the one easiest available. Aside from the hardware some software was also needed, the laptop used for the attacks had to run linux for be able to run the correct drivers and software. We used a normal debian lenny installation running the latest cvs versions of the (in)famous MadWifi-ng wireless drivers which are capable to do almost anything when it comes to 802.11 networks. Together with the MadWifi-ng drivers we ran an old, but working version of the wireless detector tool Kismet for finding out BSSIDs for the targets. To launch the actual attacks we used the latest version of the aircrack-ng suite who offers tools for attacking wireless networks in a large number of ways and it was essential in all our attacks. For making the attack against WPA with a rainbow table we used the tool called CoWPAtty and a downloaded table matching our targets ESSID. To make the Evil Twin attack a bit more amusing we ran a tool called sslstrip who maps https links to http ones, together with the softAP tool called airbase-ng from the aircrack suite. We decided to not attack the MIC in WPA mostly because the tools we have don't work with the MadWifi-ng drivers yet. Other tools we used were things like iptables and dns- and dhcp servers.

4.2 Breaking WEP

The first experiment we tried out was breaking WEP. With a bit of pre-work we where able to break it within a minute. As mention before we used aircrack-ng to intercept and inject packets. Besides the tools and hardware there are a few things you need to know to be able to break WEP this way, the BSSID for the AP we are going to attack, MAC-address for the PC running the attack, the AP channel and the wireless interface. The MAC-address for your wireless interface is already known, and to get the BSSID of the AP and what channel it is using can easily be obtained by the help of kismet. Kismet scans for all closely networks and list information about them including the BSSID and channel used. To be able to crack the WEP key we need to gather a lot of IV:s (initialization vectors). Under normal circumstances networks don't generate these IV:s very quickly, and it can take some time before you have gathered enough to break the WEP-key. Luckily we can speed up this process by using injections. Here follows a total of 4 steps we preformed to break the WEP-key.

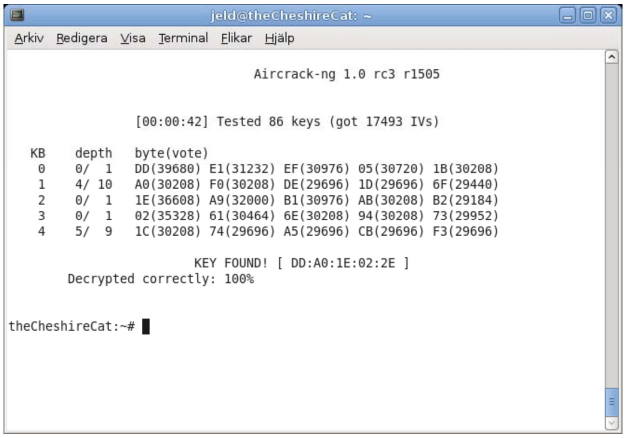
Step 1) For the packet injection to work the source MAC-address must already be associated or else the AP will ignore the sent packet. So the first thing we do is to make a fake authentication with the AP using airplay-ng.

Step 2) Here we will start listening for ARP requests, forge them and inject them back into the network forcing the AP to broadcast them again with new IV:s.

Step 3) At the same time we start injecting packets we use airdump-ng to capture all IV:s sent from the AP and saves them to a file.

Step 4) The last step is to run aircrack-ng, this can be done both offline when enough IV:s has been captured and saved to a file or online at the same time we gather IV:s.

The result of our attack can be seen in fig1. This was an online attack and as we can see it only toke 42 sec to get the key.[14]



```
                              Aircrack-ng 1.0 rc3 r1505


                 [00:00:42] Tested 86 keys (got 17493 IVs)

  KB    depth   byte(vote)
   0    0/  1   DD(39680) E1(31232) EF(30976) 05(30720) 1B(30208)
   1    4/ 10   A0(30208) F0(30208) DE(29696) 1D(29696) 6F(29440)
   2    0/  1   1E(36608) A9(32000) B1(30976) AB(30208) B2(29184)
   3    0/  1   02(35328) 61(30464) 6E(30208) 94(30208) 73(29952)
   4    5/  9   1C(30208) 74(29696) A5(29696) CB(29696) F3(29696)

                         KEY FOUND! [ DD:A0:1E:02:2E ]
        Decrypted correctly: 100%


theCheshireCat:~#
```

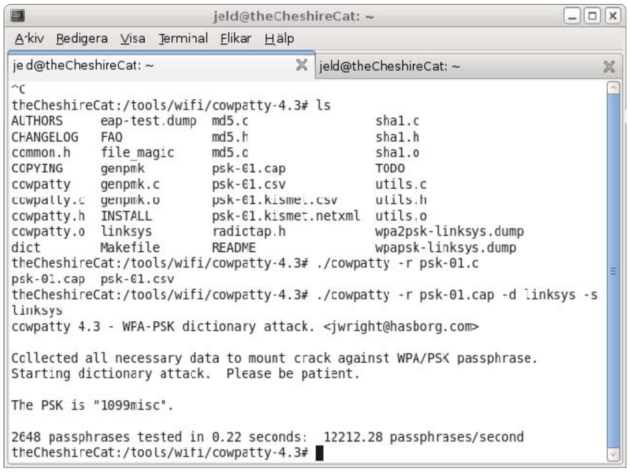
Figure 1. Result of Aircrack-ng, getting the WEP key

4.3 Deploying an Evil Twin

The next experiment was to deploy an Evil Twin. For this we used airbase-ng. To make this work there is not really much that need to be done. Airbase-ng is setup so it will respond to any prob request with a proper prob response, all you need to know is if the AP you are pretending to be is using WEP or WPA and recover the key. Another thing you can do to make the client more likely to associate with your own AP instead of the real one is to dissociate the real AP. This can be done with airplay-ng. With the Evil Twin a man-in-the-middle attack is preformed, capturing any data sent from the client. [17].

4.4 Breaking WPA

To break WPA we used a so called dictionary attack and the tool we choose to use was coWPAtty. To make this work you first need to capture the WPA four-way handshake before running coWPAtty. The handshake is captured the same way as in WEP but with a few flag changes. Normally this can take quite some time to wait for a client to connect to the AP. To speed it up we use airplay to de-authenticate the clients from the AP, forcing them to re associate with it. After getting the handshake we ran coWPAtty against a pre-computed hash. The result can be seen in fig2.



```
^C
theCheshireCat:/tools/wifi/cowpatty-4.3# ls
AUTHORS     eap-test.dump  md5.c                  sha1.c
CHANGELOG   FAQ            md5.h                  sha1.h
common.h    file_magic     md5.o                  sha1.o
COPYING     genpmk         psk-01.cap             TODO
cowpatty    genpmk.c       psk-01.csv             utils.c
cowpatty.c  genpmk.o       psk-01.kismet.csv      utils.h
cowpatty.h  INSTALL        psk-01.kismet.netxml   utils.o
cowpatty.o  linksys        radiotap.h             wpa2psk-linksys.dump
dict        Makefile       README                 wpapsk-linksys.dump
theCheshireCat:/tools/wifi/cowpatty-4.3# ./cowpatty -r psk-01.c
psk-01.cap  psk-01.csv
theCheshireCat:/tools/wifi/cowpatty-4.3# ./cowpatty -r psk-01.cap -d linksys -s
linksys
cowpatty 4.3 - WPA-PSK dictionary attack. <jwright@hasborg.com>

Collected all necessary data to mount crack against WPA/PSK passphrase.
Starting dictionary attack.  Please be patient.

The PSK is "1099misc".

2648 passphrases tested in 0.22 seconds:  12212.28 passphrases/second
theCheshireCat:/tools/wifi/cowpatty-4.3# █
```

Figure 2. Breaking WPA with coWPAtty

Instead of a pre-computed hash you could use a dictionary file and it would give the same result but take longer because you would need to hash the password and SSID before comparing.[12]

5. Conclusions

We have found out in this practical study on wireless network security that there are a lot of security flaws and they are very well documented, finding information both for the theoretical part and the experiments were easy.

During the preparation for the practical experiments we noticed that almost everything except the Evil Twin attack was described in easy step-by-step guides on several web pages, the level of knowledge needed to launch several of the attacks is scary low. We think that this is both good and bad, good in the sense that with well documented security flaws people should notice the need of better security, unfortunately it seems that a lot of people either don't know or care about this, as you still can find networks unprotected or with weak protection. Then there are the problems with the open networks whom are vulnerable to many attacks, we think that this is a problem that is hard to fix when those problems exist in probably all wireless network since the air is hard to control. However, there has been some articles of the insecurity of wireless networks in the news over past few years. With a quick scan at neighborhood it seems that such articles has given result, we found almost only WPA and WPA2 encrypted networks. If the users have chosen good passwords is another thing. In this project we have realized that to keep you wireless network well secured today you really should use WPA2 with AES CCMP and a good long password together with a strange ESSID to make sure wordlist, rainbow table and normal brute force attacks aren't easy. WPA with a good strong password is properly enough in most cases but since tools for packet injecting WPA protected networks using TKIP is out public, bad things can happen [18].

6. References

[1] Institute of Electrical and Electronic Engineers: 802.11 Standard http://standards.ieee.org/getieee802/802.11.html [2009-04-03]

[2] J. Geier, "Understanding 802.11 frame types" http://www.wi-fiplanet.com/tutorials/article.php/1447501 [2009-04-04]

[3] N. Borisov, I. Goldberg, D. Wagner, "Intercepting mobile communication: The Insecurity of 802.11", http://www.isaac.cs.berkeley.edu/isaac/mobicom.pdf, [2009-04-11]

[4] Wikipedia "Wi-Fi Protected Access", http://en.wikipedia.org/wiki/Wi-Fi_Protected_Access, [2009-04-11]

[5] D. Byers, IDA at Linköpings Universitet http://www.ida.liu.se/~TDDD17/lectures/slides/tddd17_lec03_net.pdf [2009-04-11]

[6] S. Vibhuti, "IEEE 802.11 WEP(Weird Equivalent Privacy) Concepts and Vulnerability" http://www.cs.sjsu.edu/faculty/stamp/CS265/projects/Spr05/papers/WEP.pdf [2009-04-11]

[7] A. Stubblefield, J. Ioannidis, A. Rubin, "Using the Fluhrer, Mantin, and Shamir Attack to Break WEP" http://www.isoc.org/isoc/conferences/ndss/02/papers/stubbl.pdf [2009-04-11]

[8] M. Beck, E. Tews, "Practical attacks against WEP and WPA" http://dl.aircrack-ng.org/breakingwepandwpa.pdf [2009-04-11]

[9] Wi-Fi Aliance, "Deploying a Wi-Fi Protected acess (WPA) and WPA2 in the Enterprise " http://www.wi-fi.org/files/kc/WPA-WPA2_Implementation_2-27-05v2.pdf [2009-04-11]

[10] J. van Rantwijk "WPA key calculation" http://www.xs4all.nl/~rjoris/wpapsk.html [2009-04-04]

[11] A. Stone, "The Michael Vulnerability" http://www.wi-fiplanet.com/columns/article.php/1556321 [2009-04-04]

[12] Wirelessdefens.org, http://www.wirelessdefence.org/Contents/Aircrack-ng_WinAircrack.htm [2009-04-24]

[13] G. Lehembre, "Wi-Fi Security, WEP, WPA and WPA2" http://www.hsc.fr/ressources/articles/hakin9_wifi/hakin9_wifi_EN.pdf

[2009-04-11]

[14] DarkAudax, “Simple WEP crack” http://209.85.129.132/search?q=cache:71C_KfQN1VQJ:www.aircrack-ng.org/doku.php%3Fid%3Dsimple_wep_crack+aircrack-ng+wep&cd=1&hl=sv&ct=clnk&gl=se&client=firefox-a
[2009-04-24]

[15] FR3DC3RV, “Evil Twin” http://fr3dc3rv.blogspot.com/2007/04/evil-twin.html
[2009-04-15]

[16] The shmoo group, “Rogue Squadron: Evil Twins, 802.11intel, Radical RADIUS, and Wireless Weaponry for Windows” http://airsnarf.shmoo.com/rogue_squadron.pdf
[2009-04-15]

[17] DarkAudax, “Airbase-ng” http://www.aircrack-ng.org/doku.php?id=airbase-ng&DokuWiki=d466c5a226b334d0a5e3d1950434cdd3
[2009-04-24]

[18] G. Fleishman, “Battered, but not broken: understanding the WPA crack” http://arstechnica.com/security/news/2008/11/wpa-cracked.ars
[2009-04-28]

[19] N. Cam-Winget, R. Housley, D. Wagner, and J. Walker, "Security Flaws in 802.11 Data Link Protocols" http://www.ida.liu.se/~TDDD17/literature/p35-cam_winget.pdf
[2009-05-01]

[20] darkaudax, “Tkiptun-ng” http://www.aircrack-ng.org/doku.php?id=tkiptun-ng
[2009-05-02]