WPA Migration Mode: WEP is back to haunt you...

Leandro Meiners (lmeiners@coresecurity / lmeiners@gmail.com)
Diego Sor (dsor@coresecurity.com / diegos@gmail.com)

July 2010

Abstract

Cisco access points support **WPA Migration Mode**, which enables both **WPA** and **WEP** clients to associate to an access point using the same Service Set Identifier (**SSID**). *Cisco* warns (inside a Q&A document[3]) about the dangers by stating "that security will operate at the least-secure level common to all devices" and "as a result, a passive WEP key attack could be launched against WEP users". The scenario where **WEP** clients are connected is a serious risk; besides "a passive WEP key attack", an active **WEP** cracking attack against a connected **WEP** client station (i.e. not the access point) could be launched, leveraging the **WEP** key in minutes.

We focused on analyzing the consequences of having this feature enabled when no **WEP** clients are present; for example after the migration to **WPA** has been carried out but this feature has been left enabled. According to *Cisco*'s statement we should be operating "at the least-secure level common to all devices", meaning **WPA**; however, we found that it is possible for an attacker to crack the **WEP** key under this scenario (i.e. no **WEP** clients) and connect to the network. This is accomplished by mounting an active attack against the access point with migration mode enabled (and no **WEP** clients) to recover the **WEP** key; once recovered, it is possible to connect to the access point using this key (as it is operating in **WPA Migration Mode**) and access the network.

Furthermore, *Cisco* also offers an additional security setting "broadcast key rotation" that according to the documentation[4] "in WPA Migration Mode, this feature significantly improves the security of key-management capable clients when there are no static-WEP clients associated to the access point". We also found that this setting could be trivially bypassed.

The obvious solution is to disable **WPA Migration Mode**; thus disabling support for legacy **WEP** stations. We further discuss mitigation strategies and suggest alternative configurations that support legacy **WEP** stations in a more secure manner.

Contents

1. Introduction

This paper presents an attack against *Cisco*'s **WPA Migration Mode**, which enables both **WPA** and **WEP** clients to associate to an access point using the same Service Set Identifier (**SSID**).

The paper is structured in the following manner: it begins by explaining some basics about **WEP** and the details of the attacks against **WEP** that are used to attack **WPA Migration Mode** (in 2). It continues by presenting *Cisco*'s **WPA Migration Mode**, explaining it detail how it works and how to configure an access point to support it (in 3). Then it proceeds to describe in detail how the attack works and its consequences (in 4). Finally, it presents mitigations against the attack and certain recommendations to aid in protecting against similar attacks (in 5).

2. Brief introduction to WEP

This section of the paper explains the inner workings of **WEP** (in 2.1). It then proceeds to provide a general overview of the attacks against it, and explains in detail the "bitflipping" attack (in 2.2). This attack against **WEP** is used by the **WPA Migration Mode** attack presented later in this paper.

2.1 WEP basics

WEP encryption is based on the **RC4** stream cipher. **WEP** encapsulation starts by generating an initialization vector (**IV**), which is appended to the **WEP** key, and together are called seed. Then, the integrity checksum value (**ICV**) is computed, which is implemented using the **CRC-32** algorithm, and appended to the data to be encrypted. Next, the key stream is generated by using the **RC4** algorithm with the seed as input. Encryption occurs by performing a bitwise exclusive-or of the key stream with the plaintext data (which includes the **ICV**). Finally, a frame is assembled that includes the **IV** as part of the header (in plaintext) and where the payload is the result of the encryption step. The following figure summarizes this procedure.

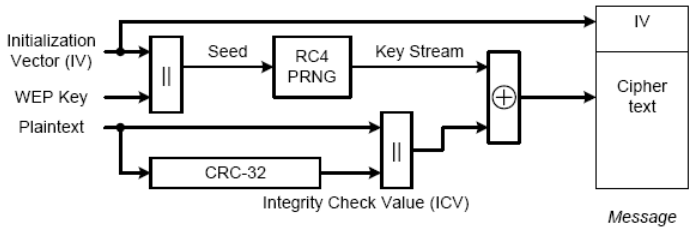

Figure 2.1: WEP encapsulation procedure

2.2 Attacks against WEP

The **WEP** protocol has been thoroughly studied and found to be riddled with weaknesses. There are two major classes of attacks: those that recover a valid keystream ([12], [6], [1]) and those that recover the encryption key ([9], [11]). These attacks leverage different weaknesses in the encryption scheme.

One of **WEP**'s weakness, is that the integrity check value uses a function (**CRC-32**) that is linear in relation to the encryption process (i.e. the exclusive-or operation). Relying on a linear function for integrity has the disadvantage that it is possible to apply a mask to the encrypted value, and compensate the integrity check value accordingly; to obtain an encrypted message that is valid and whose decrypted value is that of applying the mask to the plaintext. This technique is commonly refered to as "bitflipping" in the literature. For complete details on the mathematics on the attack see [2].

3. WPA Migration Mode

This section of the paper describes in detail how **WPA Migration Mode** works (in 3.1), then proceeds to describe how to setup this operating mode in a *Cisco* access point (in 3.2). Finally, it explains how to detect an access point that is configured to operate under this setup (in 3.3).

3.1 WPA Migration Mode technical details

Cisco's **WPA Migration Mode** allows stations that support the following types of authentication and encryption schemes, to associate to the access point using the same **SSID**:

- **WPA** clients capable of **TKIP** and authenticated key management.
- **IEEE 802.1X** compliant clients (such as legacy **LEAP** clients and clients using **TLS**) capable of authenticated key management but not **TKIP**.
- **WEP*** clients not capable of **TKIP** or authenticated key management.

This is accomplished by setting the multicast cipher suite for the **SSID** to be **WEP**, allowing **WEP** and **TKIP** stations to associate to the access point, and having the access point keep an internal state whereby it knows how to encrypt the unicast frames it must forward to each particular station.

This works due to several contributing factors:

1. The authentication-association procedure performed by **WEP** and **TKIP** stations can be distinguished[7].
2. **IEEE 802.11** networks are "switched".
3. Multicast traffic is encrypted using **WEP**.

In the case of unicast traffic, the fact that the authentication-association procedure performed by **WEP** and **TKIP** stations can be distinguished allows the access point to keep an internal state whereby it can track the encryption settings supported by each station that has joined the network. Coupled with the

*Using a static **WEP** key.

fact that **IEEE 802.11** networks are "switched", it allows the access point to forward the frames encrypted with the correct settings (be it **WEP** or **TKIP**) to each station. It is important to note that the standard behavior in a **WEP** or **TKIP** network is that the access point decrypts the frames sent by a station and encrypts it again prior to forwarding it (if the destination is a wireless station)[†]. This is what makes it possible for the access point to use a different encryption schemes for unicast traffic without the stations being aware that the access point is doing so.

In the case of multicast traffic, every station is able to "understand" it as the multicast frames are sent with **WEP**, which is the lowest common encryption mechanism supported by all stations. **WEP** stations expect this to be the case (i.e. multicast traffic being protected by **WEP**) and it is indiferent to **TKIP** stations as the **IEEE 802.11** standard supports using **WEP** as the multicast cipher suite (refer to section "7.3.2.25.1 Cipher suites" of [5]).

3.2 Configuring WPA Migration Mode

To set up an **SSID** for **WPA Migration Mode**, the following settings must be configured:

- **WPA** optional.
- A cipher suite containing **TKIP** and 40-bit or 128-bit **WEP**.
- A static **WEP** key in key slot 2 or 3.

The following example shows the *IOS* commands to sets the **SSID** "migrate" for **WPA Migration Mode**:

```
ap# configure terminal
ap(config)# interface dot11radio 0
ap(config-if)# ssid migrate
ap(config-if-ssid)# authentication open
ap(config-if-ssid)# encryption mode ciphers tkip wep128
ap(config-if)# encryption key 2 size 128 AAAAAAAAAAAAAAAAAAAAAAAA transmit-key
ap(config-if)# ssid migrate
ap(config-if-ssid)# authentication key-management wpa optional
ap(config-if-ssid)# wpa-psk ascii migrationmode
ap(config-if-ssid)# end
ap# end
```

[†]Except under certain **QoS** settings station-to-station communication is not performed; all traffic goes through the access point.

For more details on the configuration, see section “Configuring WPA Migration Mode” of “Cisco IOS Software Configuration Guide for Cisco Aironet Access Points” ([4]).

3.3 Detecting an AP with WPA Migration Mode enabled

When attempting to detect an access point that has **WPA Migration Mode** enabled, there is the obvious solution of testing for the behavior that characterizes it: allowing both **WEP** and **WPA** stations to connect. Therefore, if attempting to connect as a **WEP** station and as a **WPA** station succeeds, the access point supports **WPA Migration Mode**. This has two major drawbacks. First, if the access point has MAC address filtering enabled connection attempts could fail due to this and not because of the access point’s supported encryption schemes. Secondly, this discovery process is active, which means it could be potentially picked up by a Wireless IPS.

The passive approach to detecting if an access point has **WPA Migration Mode** enabled, relies on one of the configuration settings that it leverages so that both **WEP** and **WPA** stations can understand multicast traffic: using **WEP** as the multicast cipher suite for **WPA**. This can be gleaned from *beacon* frames by analyzing the **WPA** tag; as shown in the following screenshot of a captured *beacon* frame:

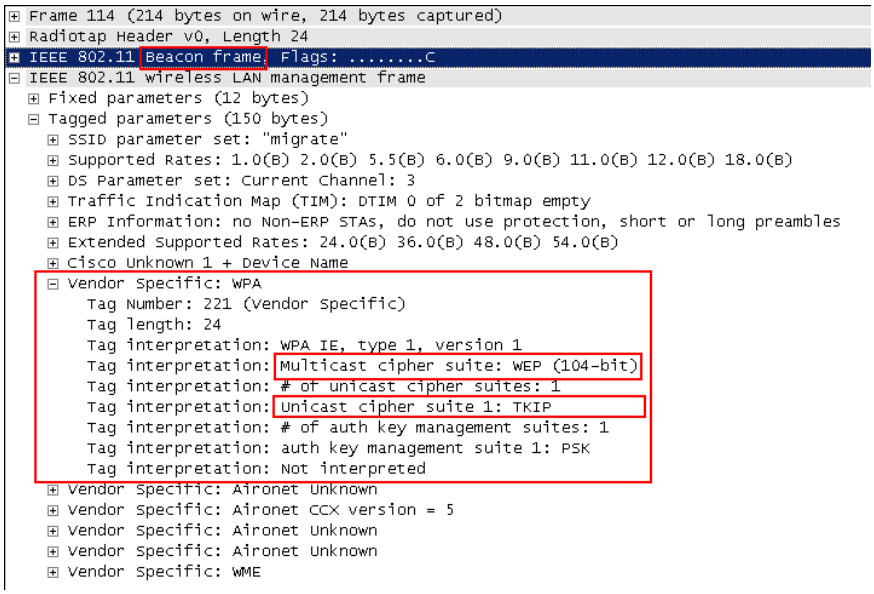

Figure 3.1: Beacon frame detailing WPA tags

This can easily be implemented as a *Wireshark* filter, by looking for frames matching the following criteria:

- Beacon frame:
  wlan.fc.type_subtype == 0x08
- With a WPA Information element:
  wlan_mgt.tag.number == 221
- Multicast cipher suite is **WEP** (40 or 104 bit):
  wlan_mgt.tag.interpretation == "Multicast cipher suite: WEP (40-bit)" or wlan_mgt.tag.interpretation == "Multicast cipher suite: WEP (104-bit)"
- Unicast cipher suite is **TKIP**:
  wlan_mgt.tag.interpretation == "Unicast cipher suite 1: TKIP"

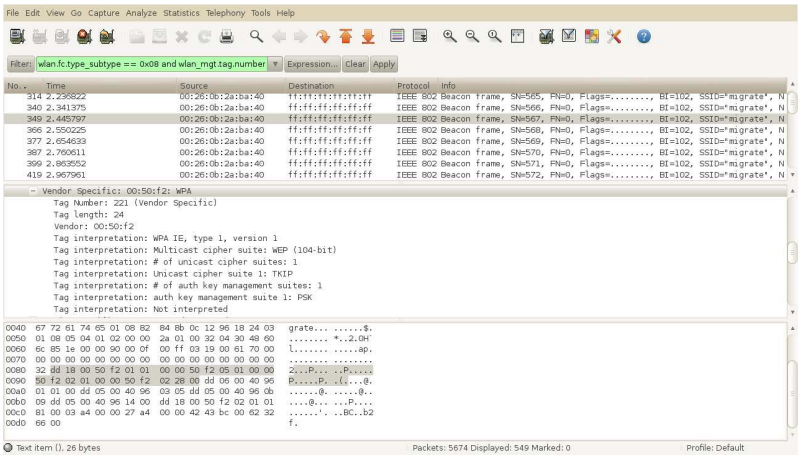

Figure 3.2: Wireshark filter to detect an AP with **WPA Migration Mode** AP enabled

4. The attack...

When considering attacking an access point with **WPA Migration Mode** enabled, there are two scenarios to consider: when **WEP** stations are still using the access point (analyzed in 4.1) and when there are no **WEP** stations in sight (for example after the migration to **WPA** has been carried out but this feature has been left enabled), considered in 4.2.

Cisco also offers an additional security setting *broadcast key rotation*, to use with **WPA Migration Mode** enabled. In 4.3 this mechanism is analyzed from a security standpoint and shown to be ineffective.

Finally, in 4.4 the steps used to join the network using the **WEP** key recovered are detailed.

4.1 WEP stations still hanging around...

The scenario where **WEP** stations are connected presents a serious risk, as it is open to classic attacks against **WEP**. *Cisco* partially warns (inside a Q&A document[3]) about these dangers by stating "that security will operate at the least-secure level common to all devices" and "as a result, a passive WEP key attack could be launched against WEP users". Besides "a passive WEP key attack", an active **WEP** cracking attack against a connected **WEP** station (i.e. not the access point) could be launched; leveraging the **WEP** key in minutes. As the target of the attack in this case is a station, the access point's configuration is not pertinent. Therefore, a standard **WEP** attack against the station can be launched successfully. The steps to carry out the attack are as follows:

1. Passively wait (and capture) for a broadcast **ARP** frame*(distinguished by its characteristic size.) that is answered by a **WEP** station.
2. Replay the captured frame.
3. Capture the **ARP** replies sent by the **WEP** station.
4. After enough frames have been captured (roughly 40.000), run *aircrack-ng* against the captured frames to obtain the **WEP** key.

*Since it is a broadcast frame it will be **WEP**-encapsulated.

This attack works because the the broadcast **ARP** frame is **WEP**-encapsulated, since it is a broadcast frame, and can therefore be replayed (**WEP** offers no replay protection). The **ARP** response (an unicast frame) is also **WEP**-encapsulated as the station is a **WEP** station. After enough **WEP**-encapsulated **ARP** responses have been gathered it is possible to crack the **WEP** key using known techniques ([9] or [11]) and readily available tools ([10]).

For complete details on how to carry out this attack using ([10]) refer to: `http://aircrack-ng.org/doku.php?id=how_to_crack_wep_via_a_wireless_client`.

4.2 No WEP stations in sight...

We found that it is possible for an attacker to crack the **WEP** key under this scenario (i.e. no **WEP** clients) and connect to the network. This is accomplished by mounting an active attack against the access point with **WPA Migration Mode** enabled (and no **WEP** clients) to recover the **WEP** key.

Since the broadcast frames are sent **WEP**-encapsulated, it is possible to crack the **WEP** key by patiently (very patiently) capturing broadcast traffic (forwarded by the access point). Knowing that this attack vector is possible the idea was to analyze how to speed up the capture processes. For this to happen, we need to be able to inject a **WEP**-encapsulated **ARP** request which elicits a **WEP**-encapsulated **ARP** response. The following procedure outlines how to accomplish this:

1. Perform an authentication and association as a **WEP** station against the target access point*.
2. Passively wait (and capture) for a broadcast **ARP** frame† (distinguished by its characteristic size.).
3. "Bitflip" the captured frame to convert it into a **ARP** request sent by the attacker station (from a random IP address).
4. Replay the "bitflipped" frame with the *To-DS* bit set.
5. Capture the **ARP** requests and replies forwarded by the access point.
6. After enough frames have been captured (roughly 40.000), run *aircrack-ng* against the captured frames to obtain the **WEP** key.

*It is not necessary for a **WEP** station to prove knowledge of the **WEP** key when *open system* authentication is in use; therefore it is possible to perform an authentication and association without knowing the key.

†Since it is a broadcast frame it will be **WEP**-encapsulated.

It is important to note that the **ARP** request forwarded by the access point is **WEP**-encapsulated as it is a broadcast frame and that the access point will forward the **ARP** reply as a **WEP**-encapsulated frame, even though it is an unicast frame, as the attacker's station joined the network as a **WEP** station.

Depending on the original **ARP** frame captured, the "bitflipped" frame might not elicit a response; because the original frame did not either. In this case, the whole procedure can be repeated until the "bitflipped" **ARP** frame captured does elicit a response, which will occur when the **ARP** frame captured elicited a response.

The attack is carried out in this manner to speed up the capture process (each frame sent generates two **WEP**-encapsulated frames); it is possible to merely replay the captured broadcast **ARP** as if the **WEP** station had sent it:

- With source MAC address changed to that of the **WEP** station.
- With the *To-DS* bit set.

It is worth mentioning that if the frame captured is replayed with the original source MAC address, the access point will not forward it; thus, not generating a new **WEP**-encapsulated frame. This happens because the station which sent the **ARP** request is a **WPA** station and therefore the broadcast frames it sends are **TKIP**-encapsulated. It is important to recall that in **TKIP** the frames sent by the station are always encrypted with the temporal encryption key shared by the station and access point, and if the frame happens to be a multicast frame it is then forwarded by the access point encrypted with the group encryption key so that all stations can decipher it.

After enough **WEP**-encapsulated **ARP** request and responses forwarded by the access point have been gathered, it is possible to crack the **WEP** key using known techniques ([9] or [11]) and readily available tools ([10]).

For our implementation of the tool visit:
`http://corelabs.coresecurity.com/index.php?module=Wiki&action=view&type=publication&name=WPA_Migration_Mode`.

4.3 Bypassing *broadcast key rotation*

Cisco offers an additional security setting *broadcast key rotation* that according to the documentation[4] "in WPA Migration Mode, this feature significantly improves the security of key-management capable clients when there are no static-WEP clients associated to the access point".

The following example shows the *IOS* commands to enable *broadcast key rotation* in **WPA Migration Mode**:

```
ap# configure terminal
```

```
ap(config)# interface dot11radio 0
ap(config)# broadcast-key change 300 capability-change
ap(config)# end
ap# end
```

Broadcast key rotation works in the following manner: "the access point generates and distributes a dynamic group key when the last non-key management (static WEP) client disassociates, and it distributes the statically configured WEP key when the first non-key management (static WEP) client authenticates".

Therefore, when no **WEP** clients are present the key in use is rotated periodically; thwarting **WEP** cracking attempts. However, this protection can be trivially bypassed by performing an authentication and association as a **WEP** station against the target access point; as per its definition the access point "distributes the statically configured WEP key when the first non-key management (static WEP) client authenticates".

Since it is not necessary for a **WEP** station to prove knowledge of the **WEP** key when *open system* authentication is in use*, there is no protection mechanism in place to prevent an attacker from triggering the key distribution mechanism to force using the static **WEP** key by simply performing an authentication-association as a **WEP** station against the target access point.

This attack can be carried out by using readily available tools ([10]), in the following manner:

```
aireplay-ng -1 0 -e <SSID> -a <AP MAC> -h <Attack MAC> <WIFI INTERFACE>
```

4.4 We have the WEP key... now what?

After the **WEP** key has been recovered using readily available tools ([10]), there is one minor caveat to take into account before connecting to the network as a **WEP** station.

Besides the standard configuration settings that must be used to connect to a **WEP** network (i.e. the **SSID** and **WEP** key), it is necessary to set the **WEP** key ID (or slot) in use.

The **WEP** scheme supports specifying up to four keys. Probably because no key management was detailed in the **IEEE 802.11** protocol standard, this was specified to aid a network administrator in rotating keys. Most access points are configured to use the first key slot.

As detailed in the explanation of configuring **WPA Migration Mode** (in 3.2), the access point is configured to use either slot 2 or 3. In order to determine

*__WPA Migration Mode__ uses *open system* authentication.

the key slot in use by the access point, it is necessary to view a captured **WEP**-encapsulated frame and determine the value of the *Key ID* field present in the frame[†]. This can be observed in figure 4.1.

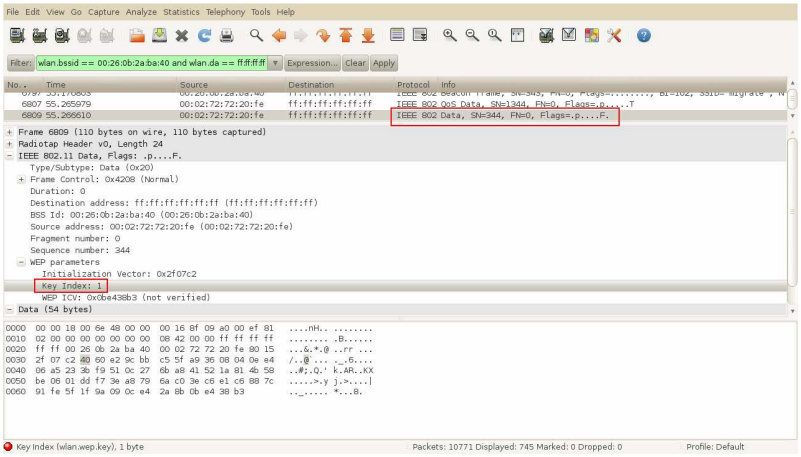

Figure 4.1: **WEP**-encapsulated frame showing *Key ID* field

Once all the configuration elements have been obtained (i.e. the **SSID**, **WEP** key and the key slot) it is possible to connect to the network by setting these value appropriately in the connecting station. After a successful connection to the network... let your imagination fly free.

It is important to note that unless other filtering mechanisms are in place, it is possible to communicate with the wireless stations and wired stations alike (the access point will take care of the appropriate encryption scheme translations to speak to **WPA** stations).

[†] *Cisco* labels key slots from one to four and the *Key ID* field goes from zero to three, therefore it is necessary to add one to the value contained in the frame to deduce the key slot.

5. Mitigations and further recommendations...

The obvious solution is to disable **WPA Migration Mode**; thus disabling support for legacy **WEP** stations. And is the recommended course of action in the case that there are only **WPA** stations using the access point.

However, if legacy **WEP** stations must be supported, this section presents different mitigation strategies and suggests alternative configurations that support them in a more secure manner.

5.1 Mitigation strategies

The following recommendations make carrying out the attack harder, but not impossible.

- Enable **PSPF** (Public Secure Packet Forwarding).*
- Enable **MAC** filtering.
- Limit signal strength (to only cover the required area).
- Implement time-based access control.

Enabling **PSPF** (a.k.a. AP/client isolation), will prevent the attack from being carried out against a wireless station, but it will not prevent replaying an **ARP** frame to a station on the wired side. Furthermore, filtering **ARP** traffic between the wired and wireless networks will aid in thwarting our implementation of the attack; however, it will still be possible to carry out the attack using frames other than **ARP**.

5.2 Alternative configurations to WPA Migration Mode

A better approach would be to use separate *VLANs* for **WEP** and **WPA** stations and disabling **WPA Migration Mode**; however, this requires changing the con-

*__PSPF__ prevents client devices associated to an access point from exchanging unicast, broadcast, or multicast traffic.

figuration of either **WEP** or **WPA** stations as each *VLAN* must have a different **SSID**.

The advantage to this approach is that more stringent layer two and three access controls can be placed on the **WEP** station *VLAN*. For more recommendations on taking this approach refer to “Integrated deployments” in [8].

If the **WEP** network has static **ARP** entries, **ARP** traffic is filtered between wired and wireless sides, **MAC** filtering is enabled, and **PSPF** is enabled, standard attacks against **WEP** are much harder to execute. Furthermore, since it is known that the **WEP** key can eventually be cracked, the idea is to limit extent of a possible breach by severely limiting the systems to which a **WEP** station is allowed to connect or by using an encryption solution (such as a VPN) over the **WEP** network. Also, deploying a wireless *IPS* can aid in detecting attacks.

6. Conclusion

This paper presented an active attack against the access point with **WPA Migration Mode** enabled to recover the **WEP** key; both when there are **WEP** stations present (trivial case) and when there are only **WPA** stations using the access point.

We strongly suggest disabling **WPA Migration Mode** if there are no **WEP** stations using the access point, as having this feature enabled lowers the security provided by **WPA** to that provided by **WEP**; thus allowing an attacker to gain access to the wireless network.

If **WEP** stations are still in use, we recommend segmenting them into a separate wireless network with very stringent security filters in place (at layer two and three), as a **WEP** network does not offer any determent to a determined attacker. Furthermore, we also urge using an encryption solution (such as a VPN) over the **WEP** network.

Bibliography

[1] Andrea Bittau, Mark Handley, and Joshua Lackey. The final nail in WEP's coffin. In *IEEE Symposium on Security and Privacy*, pages 386–400. IEEE Computer Society, 2006. ISBN 0-7695-2574-1. URL http://doi.ieeecomputersociety.org/10.1109/SP.2006.40. [cited at p. 2]

[2] Nikita Borisov, Ian Goldberg, and David Wagner. Intercepting mobile communications: the insecurity of 802.11. In *MOBICOM*, pages 180–189, 2001. URL http://doi.acm.org/10.1145/381677.381695. [cited at p. 3]

[3] Inc. Cisco Systems. *WI-FI PROTECTED ACCESS, WPA2 AND IEEE 802.11I Q&A*, 2004. URL http://www.cisco.com/en/US/customer/netsol/ns339/ns395/ns176/ns178/netqa0900aecd801e3e59.html. [cited at p. i, 8]

[4] Inc. Cisco Systems. *Cisco IOS Software Configuration Guide for Cisco Aironet Access Points*, 2010. URL http://www.cisco.com/en/US/docs/wireless/access_point/12.4_10b_JA/configuration/guide/scg12410b.html. [cited at p. i, 6, 10]

[5] IEEE. *IEEE Std 802.11i, Amendment to IEEE Std 802.11 - Amendment 6: Medium Access Control (MAC) Security Enhancements*. IEEE, 2004. [cited at p. 5]

[6] Korek. chopchop (experimental wep attacks), 2004. URL http://www.netstumbler.org/showthread.php?t=12489. [cited at p. 2]

[7] Prabhash Dhyani MD Sohail Ahmad. Wi-fish finder: Who will bite the bait. DEF CON Communications, 2009. URL http://www.defcon.org/images/defcon-17/dc-17-presentations/defcon-17-md_sohail_ahmad-wi-fish.pdf. [cited at p. 4]

[8] Balinsky A. Sankar k., Sundaralingam S. *Cisco Wireless LAN Security*, 2005. [cited at p. 14]

[9] Adam Stubblefield, John Ioannidis, and Aviel D. Rubin. Using the fluhrer, mantin, and shamir attack to break WEP. In *NDSS*. The Internet Society, 2002. ISBN 1-891562-14-2; 1-891562-13-4.

URL http://www.isoc.org/isoc/conferences/ndss/02/proceedings/papers/stubbl.pdf. [cited at p. 2, 9, 10]

[10] The Aircrack-NG team. Aircrack-ng suite. URL http://www.aircrack-ng.org. [cited at p. 9, 10, 11]

[11] Erik Tews, Ralf-Philipp Weinmann, and Andrei Pyshkin. Breaking 104 bit wep in less than 60 seconds. Cryptology ePrint Archive, Report 2007/120, 2007. URL http://eprint.iacr.org/2007/120.pdf. [cited at p. 2, 9, 10]

[12] Y.C.J. Wav W.A. Arbaugh, N. Shankar. An inductive chosen plaintext attack against wep/wep2, 2001. [cited at p. 2]