

# Wireless Infidelity II: Airjacking

Assessing the extent of the security risks involved in wireless networking technology by considering three possible scenarios demonstrating vulnerabilities.

In the previous column (September 2004), I discussed the history and practice of war driving. I noted the inherent insecurities in the 802.11 protocol standards and reported some rather frightening statistics (for example, over 60% of the Wireless Access Points discovered in the 4th Annual WorldWide War Driving Competition had no form of encryption enabled!).

I also pointed out that there is a perfectly lawful and legitimate use for wireless monitoring, but when it is used with unlawful or unethical intent, it is generally characterized as war driving. I observed that war driving is now ubiquitous: a good illustration of this is provided by the WiGLE.net online database of WAPs.

Having established that the practice of war driving is commonplace, the extent of the risk remains to be shown. We will consider three possible scenarios

here. But first, we address the necessary preliminaries.

## Service Set IDs

Since our goal is to discuss wireless security, we'll frame our discussion in terms of a very

high-level overview of wireless technology.

We normally associate the term wireless networks with the 802.11 family of protocols, the most popular of which is the various flavors of 802.11 (aka WiFi). The foundation of an 802.11 net-

work is the basic service set (BSS). Service sets may be defined as a tier structure:

Tier 0: Independent BSS (IBSS) or ad hoc network. Any cluster of wireless-enabled computers (aka stations) intercommunicating between themselves.

Tier 1: Infrastructure BSS. A cluster of one or more stations connected with a Wireless Access Point (WAP, or simply AP). In this mode, all stations communicate with each other through one WAP at a time—no direct station-to-station communication is permitted.

Tier 2: Extended Service Set (ESS). A cluster of BSSs where interconnected WAPs serve as bridges between service areas.

The Service Set ID (SSID) is a 32 byte or less network name of a service set (a list of default SSIDs

is available at [www.cirt.net/cgi-bin/ssids.pl](http://www.cirt.net/cgi-bin/ssids.pl)). This name is used by other network devices to initiate a connection. WAPs may be configured as “open” or “closed.” In the open mode, the WAP broadcasts its SSID to the world; in closed mode, it does not. A computer with a WiFi card set to SSID=ANY will attempt to authenticate with the open WAPs with the strongest signals. This is called association polling and is built into XP by default when wireless is enabled, as confirmed by the menu bar pop-up caption shown in Figure 1.

Therein is our first security consideration. Is it advisable to broadcast the name of the WAP to the world? Where WAPs are concerned, the best practice is to avoid drawing any more attention to the WAP than necessary. Disabling SSID broadcasting and setting the signal strength as low as possible without losing the signal is a good first step.

However, closed WAPs only deter primitive network beacon sniffers (for example, NetStumbler). Beacon Sniffers (aka active sniffers) continuously broadcast probe requests to entice WAPs to respond. Closed WAPs will not respond unless the probe requests contain its SSID (which means it must be known in advance), so beacon sniffers are both extremely noisy (and trivial to detect) and provide an incomplete scan. However, greater stealth can be achieved by “passive” sniffers that

operate with the network card in monitor mode. Monitor mode captures all traffic on a frequency



Figure 1. Windows XP menu bar caption indicating enabled wireless connectivity.

regardless of source or destination as long as the signal strength is adequate. This is to be distinguished from promiscuous mode, which captures all traffic on the network to which you are associated and is not a default option on all wireless cards.

In monitor mode, passive sniffers like AirMagnet and Kismet monitor all wireless transmissions close enough to detect, irrespective of source and destination, without generating any betraying traffic themselves.

So what does a closed WAP buy us? Not much, for the serious invader. But shutting off the SSID broadcast is still worth the effort, if for no other reason than it discourages casual sniffing and WAP mapping by would-be hackers.

## WEP

The goal of Wired Equivalent Privacy (WEP) was to bring some of the security available in wired networks to WiFi. Unfortunately, the designers bungled

the job (see [citeseer.ist.psu.edu/fluhrer01weaknesses.html](http://citeseer.ist.psu.edu/fluhrer01weaknesses.html)). WEP suffers from two fundamental deficiencies: it was poorly designed and it was poorly implemented. Other than that, it's fine.

A key WEP vulnerability results from the implementation of the RC4 symmetric stream cipher algorithm. Simple stream ciphers work by XORing a stream of bits (the key) with the plaintext to come up with the cipher text that is transmitted and reversed at the other end. In its simplest form, this stream cipher wouldn't be secure because a string of zeros in the plaintext would produce the actual key in the cipher text due to the way XOR works. The RC4 algorithm relies on a pseudo-random number generated initialization vector (IV) to control the scrambling of the keystream to provide the desired robustness.

The WEP implementation of RC4 is flawed in several ways, which allows the algorithm itself to be attacked and the key to be revealed. The first problem with WEP is that the IV is always prepended to the key prior to generation of the keystream by the RC4 algorithm. Secondly, the IV is relatively small (3 bytes), which produces a lot of repetitions as the scant 16.77 million variations are reused to encrypt millions of packets. Third, some of the IVs are “weak” in the sense they may be used to betray information about the key.



2a.

few million packets generate enough weak IV traffic to recover 40-bit WEP keys.

### WAPs, WEP, SSIDs, and the Art of Airjacking

We illustrate the vulnerabilities of WiFi technology with the following three example scenarios.

**Case 1: WAP with SSID broadcast enabled, and no WEP enabled.** This configuration invites the greatest vulnerability. It is also the most common, since most WAPs are shrinkwrapped with this configuration. The 2004 World Wide Wardrive Competition reported

that 27.5% of all WAPs can be placed in this category.

We begin the penetration test with a wireless scan of our environment. Figure 2a reveals the result of a scan of our lab with a WAP set up in this configuration. To distinguish it from other WAPs, we set our SSID to “NoWepWap.”

The AirMagnet screenshot shown in Figure 2a reveals a cornucopia of useful information about

When the first data to be encrypted in a WEP packet is the SNAP header (as with IP and ARP packets), the first byte of this header is almost always 0xAA. A weak IV has a format of B+3::ff:X (where B is the byte of the key to be found, ff is the constant 255, and X is irrelevant). WEP cracking usually relies on accumulated traffic produced by weak IV values. Since the IV is transmitted with the packet in plaintext, weak IVs are easy to detect. The key value of B is determined after the B+4th iteration of the key scheduling algorithm. Given a sufficient amount of traffic and repeated applications of this strategy, we can recover the entire key. As a rule of thumb, a



2b.

Figure 2a. (top) AirMagnet scan of active WAPs. Figure 2b. (bottom) Using the Windows GUI to automatically connect to an “Open” WAP. Click “connect” and you’re in.

our WAP. The top half of the left panel is the alarm window. Our WAP is identified by the box in the ninth line. We note that we’re broadcasting on channel six on