Digital Village

Hal Berghel and Jacob Uecker

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 network 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.

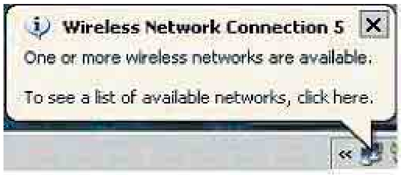
The Service Set ID (SSID) is a 32 byte or less network name of a service set (a list of default SSIDs

BEATA SZPURA

is available at 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 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.


Wireless Network Connection 5

One or more wireless networks are available.

To see a list of available networks, click here.

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

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.

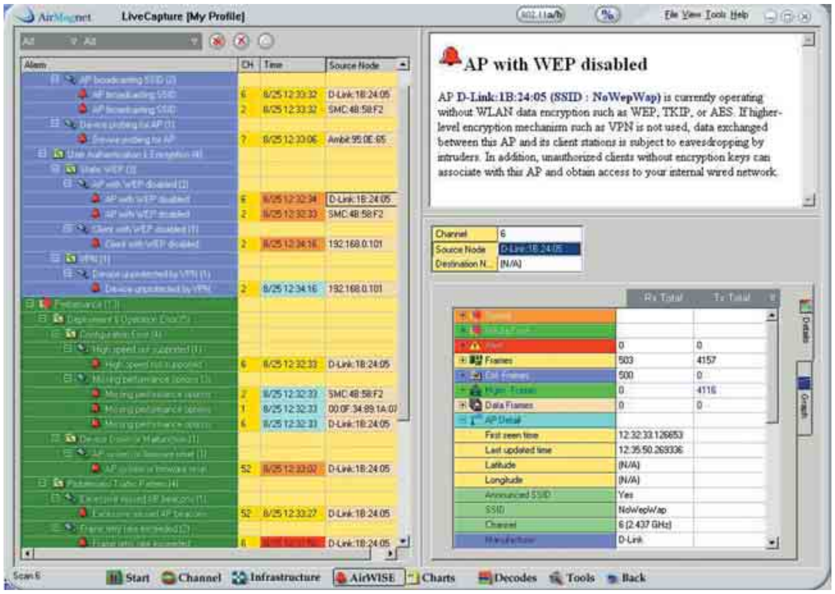
WEP

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). 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.



AP with WEP disabled

AP **D-Link:1B:24:05 (SSID : NoWepWap)** is currently operating without WLAN data encryption such as WEP, TKIP, or AES. If higher-level encryption mechanism such as VPN is not used, data exchanged between this AP and its client stations is subject to eavesdropping by intruders. In addition, unauthorized clients without encryption keys can associate with this AP and obtain access to your internal wired network.

Channel	6
Source Node	D-Link:1B:24:05
Destination N...	(N/A)

	Rx Total	Tx Total
	0	0
Frames	503	4157
	500	0
	0	4116
Data Frames	0	0
AP Detail		
First seen time	12:32:33.126653	
Last updated time	12:35:50.269336	
Latitude	(N/A)	
Longitude	(N/A)	
Announced SSID	Yes	
SSID	NoWepWap	
Channel	6 (2.437 GHz)	
Manufacturer	D-Link	

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

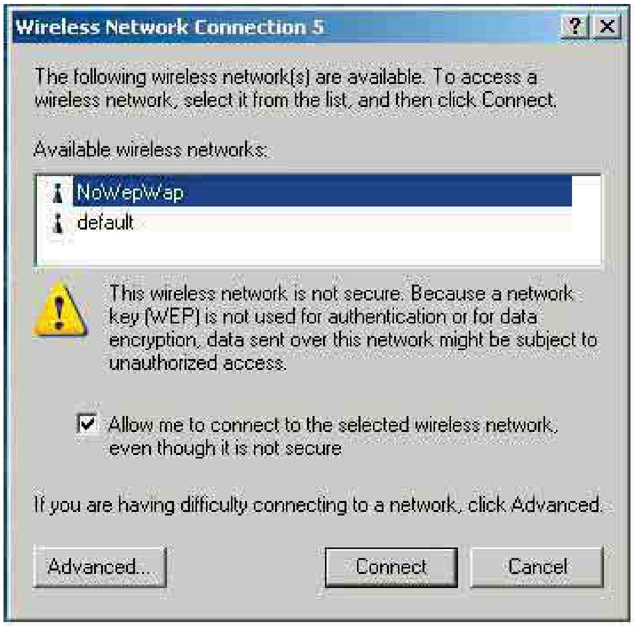
WAPS, WEP, SSIDs, and the Art of Airjacking

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 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



Wireless Network Connection 5

The following wireless network(s) are available. To access a wireless network, select it from the list, and then click Connect.

Available wireless networks:

NoWepWap
default

This wireless network is not secure. Because a network key (WEP) is not used for authentication or for data encryption, data sent over this network might be subject to unauthorized access.

☑ Allow me to connect to the selected wireless network, even though it is not secure

If you are having difficulty connecting to a network, click Advanced.

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.

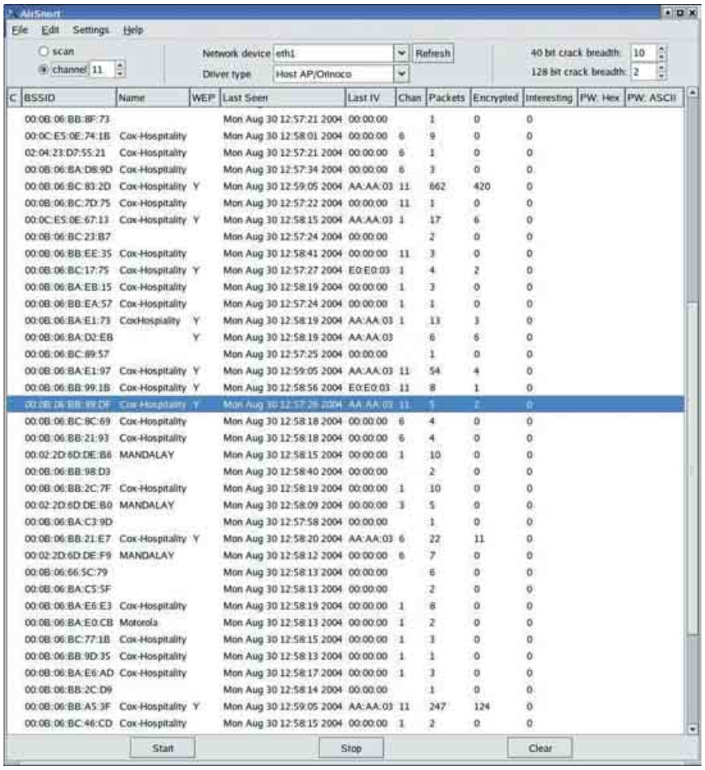
802.11g, that the manufacturer is D-Link, and the MAC address is available. There is also a red bell alarm associated with this NoWepWap that indicates we are not using encryption. (To call AirMagnet feature-rich is an understatement!) Therefore, the challenge of eavesdropping on our WAP becomes irresistible.

The connection to NoWepWap is very simple: after receiving the menu bar pop-up shown in Figure 1, we're one click away from connecting (see Figure 2b).

Case 2: WAP with SSID broadcast disabled, and no WEP enabled. This configuration offers the second greatest vulnerability level, not far behind Case 1. It is only minimally better protected because disabling SSID broadcast only hides from the primitive beacon sniffers, not the sophisticated passive scanners like Kismet and AirMagnet. It will also not appear as an available network to Windows, because Windows requires that the SSID of the target WAP is specified prior to the authentication request. If the SSID isn't known, it can't be entered manually and connectivity is blocked. While no deterrent of significance, disabling SSID broadcast might discourage your neighbor from using your wireless network to gain free Web access.

Case 3: WAP with SSID broadcast disabled, and WEP enabled. This configuration is the least vulnerable (do not read

3a.

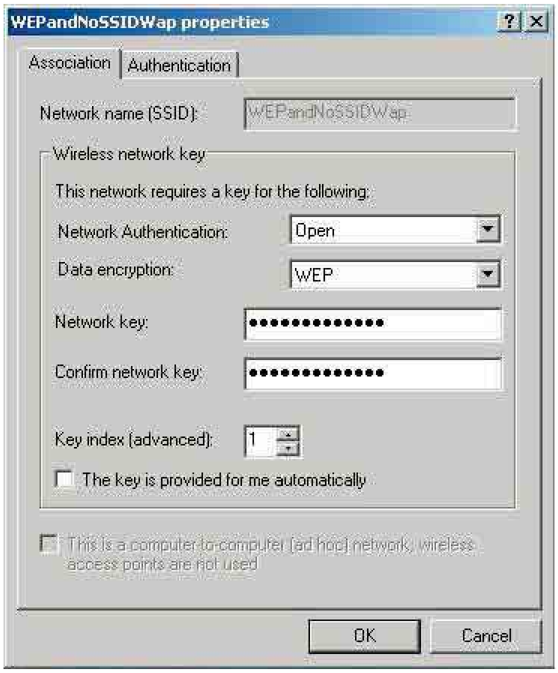


○ scan
● channel 11

Network device eth1
Driver type Host AP/Orinoco

40 bit crack breadth: 10
128 bit crack breadth: 2

C	BSSID	Name	WEP	Last Seen	Last IV	Chan	Packets	Encrypted	Interesting	PW: Hex	PW: ASCII
	00:0B:06:BB:8F:73			Mon Aug 30 12:57:21 2004	00:00:00		1	0	0		
	00:0C:E5:0E:74:1B	Cox-Hospitality		Mon Aug 30 12:58:01 2004	00:00:00	6	9	0	0		
	02:04:23:D7:55:21	Cox-Hospitality		Mon Aug 30 12:57:21 2004	00:00:00	6	1	0	0		
	00:0B:06:BA:D8:9D	Cox-Hospitality		Mon Aug 30 12:57:34 2004	00:00:00	6	3	0	0		
	00:0B:06:BC:83:2D	Cox-Hospitality	Y	Mon Aug 30 12:59:05 2004	AA:AA:03	11	662	420	0		
	00:0B:06:BC:7D:75	Cox-Hospitality		Mon Aug 30 12:57:22 2004	00:00:00	11	1	0	0		
	00:0C:E5:0E:67:13	Cox-Hospitality	Y	Mon Aug 30 12:58:15 2004	AA:AA:03	1	17	6	0		
	00:0B:06:BC:23:B7			Mon Aug 30 12:57:24 2004	00:00:00		2	0	0		
	00:0B:06:BB:EE:35	Cox-Hospitality		Mon Aug 30 12:58:41 2004	00:00:00	11	3	0	0		
	00:0B:06:BC:17:75	Cox-Hospitality	Y	Mon Aug 30 12:57:27 2004	E0:E0:03	1	4	2	0		
	00:0B:06:BA:EB:15	Cox-Hospitality		Mon Aug 30 12:58:19 2004	00:00:00	1	3	0	0		
	00:0B:06:BB:EA:57	Cox-Hospitality		Mon Aug 30 12:57:24 2004	00:00:00	1	1	0	0		
	00:0B:06:BA:E1:73	CoxHospiality	Y	Mon Aug 30 12:58:19 2004	AA:AA:03	1	13	3	0		
	00:0B:06:BA:D2:EB		Y	Mon Aug 30 12:58:19 2004	AA:AA:03		6	6	0		
	00:0B:06:BC:89:57			Mon Aug 30 12:57:25 2004	00:00:00		1	0	0		
	00:0B:06:BA:E1:97	Cox-Hospitality	Y	Mon Aug 30 12:59:05 2004	AA:AA:03	11	54	4	0		
	00:0B:06:BB:99:1B	Cox-Hospitality	Y	Mon Aug 30 12:58:56 2004	E0:E0:03	11	8	1	0		
	00:0B:06:BB:99:DF	Cox-Hospitality	Y	Mon Aug 30 12:57:26 2004	AA:AA:03	11	5	2	0		
	00:0B:06:BC:8C:69	Cox-Hospitality		Mon Aug 30 12:58:18 2004	00:00:00	6	4	0	0		
	00:0B:06:BB:21:93	Cox-Hospitality		Mon Aug 30 12:58:18 2004	00:00:00	6	4	0	0		
	00:02:2D:6D:DE:B6	MANDALAY		Mon Aug 30 12:58:15 2004	00:00:00	1	10	0	0		
	00:0B:06:BB:98:D3			Mon Aug 30 12:58:40 2004	00:00:00		2	0	0		
	00:0B:06:BB:2C:7F	Cox-Hospitality		Mon Aug 30 12:58:19 2004	00:00:00	1	10	0	0		
	00:02:2D:6D:DE:B0	MANDALAY		Mon Aug 30 12:58:09 2004	00:00:00	3	5	0	0		
	00:0B:06:BA:C3:9D			Mon Aug 30 12:57:58 2004	00:00:00		1	0	0		
	00:0B:06:BB:21:E7	Cox-Hospitality	Y	Mon Aug 30 12:58:20 2004	AA:AA:03	6	22	11	0		
	00:02:2D:6D:DE:F9	MANDALAY		Mon Aug 30 12:58:12 2004	00:00:00	6	7	0	0		
	00:0B:06:66:5C:79			Mon Aug 30 12:58:13 2004	00:00:00		6	0	0		
	00:0B:06:BA:C5:5F			Mon Aug 30 12:58:13 2004	00:00:00		2	0	0		
	00:0B:06:BA:E6:E3	Cox-Hospitality		Mon Aug 30 12:58:19 2004	00:00:00	1	8	0	0		
	00:0B:06:BA:E0:CB	Motorola		Mon Aug 30 12:58:13 2004	00:00:00	1	2	0	0		
	00:0B:06:BC:77:1B	Cox-Hospitality		Mon Aug 30 12:58:15 2004	00:00:00	1	3	0	0		
	00:0B:06:BB:9D:35	Cox-Hospitality		Mon Aug 30 12:58:13 2004	00:00:00	1	1	0	0		
	00:0B:06:BA:E6:AD	Cox-Hospitality		Mon Aug 30 12:58:17 2004	00:00:00	1	3	0	0		
	00:0B:06:BB:2C:D9			Mon Aug 30 12:58:14 2004	00:00:00		1	0	0		
	00:0B:06:BB:A5:3F	Cox-Hospitality	Y	Mon Aug 30 12:59:05 2004	AA:AA:03	11	247	124	0		
	00:0B:06:BC:46:CD	Cox-Hospitality		Mon Aug 30 12:58:15 2004	00:00:00	1	2	0	0		



Network name (SSID): WEPandNoSSIDWap

Wireless network key

This network requires a key for the following:

Network Authentication: Open

Data encryption: WEP

Network key: ••••••••••••••

Confirm network key: ••••••••••••••

Key index (advanced): 1

□ The key is provided for me automatically

□ This is a computer-to-computer (ad hoc) network; wireless access points are not used

3b.

Figure 3a. The AirSnort WEP cracker interface (note "interesting" column). Figure 3b. The Windows manual interface for authenticating to a WAP.

"secure" into this) of all of the configuration options. The 2004 World Wide Wardrive Competition reported that only 38.4% of all WAPs are in this category.

By enabling WEP and disabling the SSID broadcast, not only will the hacker have to find the SSID of the network, but also the WEP key must be found. The SSID may be obtained by adding a passive sniffer to our arsenal, so we may safely assume we have that in hand. But what do we do about the WEP barrier?

The brute force of 40-bit or 104-bit keys is unrealistic. However, we know enough about the WEP implementation of RC4 that we are confident we can count on packets that were encrypted with weak initialization vectors to give up the goods. If only there were some utility…Enter AirSnort.

Figure 3a shows the basic interface to the AirSnort WEP cracking utility. We call your attention to AirSnort's ability to determine whether packets are encrypted and, more importantly, whether they are "interesting." In this case, an interesting packet is one that was generated with a weak IV! Given enough time and volume of wireless data, AirSnort will crack any WEP key. As a rule, once AirSnort receives

URL Pearls

AirMagnet (www.airmagnet.com) is arguably the best-of-breed wireless monitor/scanner. It is a commercial product designed for Windows notebooks and PDAs. Kismet (www.kismetwireless.net), though less fully featured, is a very powerful, open source *nix utility that integrates well with AirSnort and Ethereal for weapons-grade packet analyses.

NetStumbler (www.stumbler.net) is one of the most widespread active wireless sniffers, though the volume of "noise" it produces and the fact that it will only work with "open" WAPs drastically limits its use in real-world applications.

WEP cracking tools are abundantly available. Airsnort (airsnort.shmoo.com) is an open source *nix utility, as is WepCrack (wepcrack.sourceforge.net). WepAttack (sourceforge.net/projects/wepattack) is a newer tool from Sourceforge that uses active dictionary attacks in much the same way as modern password crackers. Sourceforge's WEPWedgie (sourceforge.net/projects/wepwedgie) falls on the invasive side of the WEP-cracking wall. As Sourceforge puts it, "WEPWedgie is a toolkit for determining 802.11 WEP keystreams and injecting traffic with known keystreams."

The motherload of WAP maps is available on the Wireless Geographic Logging Engine Web site (wigle.net). Circa late September 2004, WiGLE's database and mapping technology included over 1.6 million WAPS. If you can't find the WAP of interest there, you can probably live without it. The statistics used in this column are reported on the 2004 World-Wide Wardrive Web site at www.worldwidewardrive.org.

The term WiFi is associated with the Wireless Ethernet Compatibility Alliance (WECA). Additional details are available at www.wi-fi.org (include the hyphen in the URL).

The seminal paper that started the world of WEP cracking was "Weaknesses in the Key Scheduling Algorithm of RC4," by Itsik Mantin, Adi Shamir, and Scott Fluhrer (see citeseer.ist.psu.edu/fluhrer01weaknesses.html).

A variety of WiFi security vendor sites provide additional detail on vulnerabilities. Airdefense (www.airdefense.net) and AirMagnet (www.airmagnet.com) provide several interesting white papers.

A useful guide to 802.11 wireless technology is Matthew Gast's *802.11 Wireless Networks: The Definitive Guide*, O'Reilly & Associates, Sebastopol (2002). A good introduction to WiFi security issues is chapter seven of O'Reilly's *Wireless Hacks* by Rob Flickenger (2003). Jeff Duntemann's *Wi-Fi Guide*, Paraglyph Press, Scottsdale (2004), is a useful introduction to the practical side of setting up and securing a wireless network. C a few million packets, it has enough data to recover the key.

Once this information has been found, Windows can immediately authenticate to a WAP (see Figure 3b).

Conclusion

We promised you a three-part analysis of WAP vulnerability. You should come away from this column with the thought that even the most "secure" WAP configuration is insecure. That doesn't diminish the utility of WiFi, but it should alert us to potential risks.

Actually, the dangers are far greater than stated here. We have only discussed the use of widely available utilities to authenticate to a WAP without permission. More aggressive attacks of wireless networks include denial-of-service attacks, man-in-the-middle attacks, forced deauthentication of authorized users, WAP MAC address spoofing, to name but a few. Recently, questions have been raised about the reliability of the successors to WEP, WPA, EAP, and LEAP. Wireless neighborhoods are only as safe as the neighbors. C

Hal Berghel (www.acm.org/hlb) is a professor and the director of the University of Nevada at Las Vegas School of Computer Science, and director of the University's Center for Cybermedia Research and co-director of the National Identity Theft and Financial Fraud Research and Operations Center.

Jacob Uecker (jacob@juecker.net) is a research assistant at the University of Nevada at Las Vegas Center for Cybermedia Research and the National Identity Theft and Financial Fraud Research and Operations Center.