

## Chapter 10

# Still at War

---

### *In This Chapter*

- ▶ Installing and using Kismet
  - ▶ Encrypting frames
  - ▶ Looking at WEP problems
  - ▶ Upgrading to WPA
  - ▶ Using AES
  - ▶ Using VPN
- 

**S**un Tzu wrote *The Art of War* two and a half thousand years ago. It's a simple book, but a profound one that every security professional should study. Sun Tzu wrote: "If you know the enemy and know yourself, you need not fear the result of a hundred battles." In this chapter, we show you how to learn more about your network by showing you the information an outsider can easily obtain. Then it's up to you to do something. If you know yourself and your enemy, then you should not fear the result of anyone using the tools in this chapter.

Sometimes one tool is insufficient for your needs, and you need to supplement the tool with another. The tools in this chapter help you do a better job of network discovery.

## *Using Advanced Wardriving Software*

We discuss how to use NetStumbler in the previous chapter. NetStumbler is a great tool — preferred by nine out of ten wardrivers — but it just doesn't give you everything you need. Additional applications like Kismet, Wellenreiter, and MiniStumbler provide features NetStumbler can't provide. For instance, NetStumbler does not tell you about "closed" systems or systems that don't broadcast their SSID, but Kismet does. NetStumbler is a simple beacon scanner, but Kismet is a *passive network scanner*, capable of detecting traffic from access points and clients. Also, you have to run NetStumbler on a laptop, portable, or luggable computer. But those devices are not *really* portable.

## 156 Part III: Advanced Wi-Fi Hacks

(Trust us — we spend a lot of time on the road lugging laptops around. They get *heavy*.) So we show you how to use MiniStumbler, which runs on a handheld. The following sections give you all the details.

### *Installing and using Kismet*

If you believe your destiny is to discover wireless networks, then Kismet is for you. Kismet is freeware 802.11b and g (and 802.11a with the right card) wardriving software. Kismet can capture data from multiple packet sources and can log in ethereal-, tcpdump-, and AirSnort-compatible log files. In addition, Kismet can do the following:

- ✓ Detect other scanning programs like NetStumbler
- ✓ Channel hop
- ✓ Highlight the detected default access point configurations
- ✓ Discover “closed,” “hidden,” or “cloaked” SSIDs for access points where SSID broadcast is disabled
- ✓ Identify the manufacturers of discovered access points
- ✓ Group and custom name SSIDs
- ✓ Detect Cisco products by using CDP
- ✓ Detect IP block
- ✓ Passively monitor and record wireless network data packets, including encrypted ones
- ✓ Map access point locations using a GPS
- ✓ Work with ethereal and AirSnort

Kismet runs on most UNIX-like systems, including Linux, Mac OS, and Cygwin, and supports Hermes and Prism2 chipset cards with linux-wlan-ng drivers. You can find information at the following Web sites:

- ✓ You can find more about drivers at Jean Tourrilhes’ Web page:

[www.hpl.hp.com/personal/Jean\\_Tourrilhes/Linux/Wireless.html](http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Wireless.html)

- ✓ Mark Mathew’s AbsoluteValue Systems Web page offers information about drivers as well.

[www.linux-wlan.com/linux-wlan](http://www.linux-wlan.com/linux-wlan)

- ✓ If you feel adventurous, you can learn how to install Kismet on Cygwin:

[www.renderlab.net/projects/wardrive/wrt54g/kismetonwindows.html](http://www.renderlab.net/projects/wardrive/wrt54g/kismetonwindows.html)

## Does my card support monitor mode?

You can determine whether your wireless interface supports monitor mode with your current drivers with one easy Linux command. Use the `iwpriv eth1` (or `wlan0` or whatever segment) command as root. This command shows

you any potential driver options that your card loads when Linux boots. If you don't see monitor mode, you need to find and install the applicable driver patch.

You can find Kismet at [www.kismetwireless.net](http://www.kismetwireless.net). You also can get Kismet for handheld computers — that is, iPaq/ARM and Zaurus/ARM — with embedded Linux. You need the ARM version from [www.kismetwireless.net/download.shtml](http://www.kismetwireless.net/download.shtml).

### *Preparing to install Kismet*

Before you install Kismet, you need to determine whether your wireless interface supports monitor mode. If it doesn't, you need to set it up so that it does; otherwise, you cannot use Kismet. Kismet even supports ar5k-based 802.11a cards.

If you have more than one wireless card, you can split the work of network scanning over the cards. The Kismet documentation provides information on this feature.

To get the most out of Kismet, you may want to make sure you have the following before getting started:

- ✓ **libpcap** ([www.tcpdump.org](http://www.tcpdump.org)): libpcap is a freeware program that facilitates the capturing and formatting of the frames. Kismet requires libpcap. Make sure you get a version that supports wireless sniffing.
- ✓ **ethereal** ([www.ethereal.org](http://www.ethereal.org)): ethereal is the gold standard for Linux sniffing. It's not required, but is highly recommended that you use ethereal to analyze the capture files. We discuss ethereal in Chapter 8.
- ✓ **GpsDrive** ([www.kraftvoll.at/software](http://www.kraftvoll.at/software)): GpsDrive is beggarware that provides GPS mapping. You can link Kismet to your GPS with this program.

Go ahead and download Kismet, and we'll explain how to install and run it and interpret the results.

### *Installing Kismet*

The first step to installing Kismet is configuring it by using the `configure` script. Table 10-1 shows Kismet's configuration options. To adjust an option,

## 158 Part III: Advanced Wi-Fi Hacks

append it to the `./configure` command. The following command, for example, shows you how to use the first option:

```
./configure --disable-curses
```

This is the proper way to run this script — from the current directory (although you can specify the whole path to execute the path). In Table 10-1, you see that this command disables the curses user interface.

<b>Table 10-1</b>		<b>Kismet Switches</b>	
<i>Option Description</i>		<i>Option Flag</i>	
Disable the curses user interfaces		<code>disable-curses</code>	
Disable ncurses panel extensions		<code>disable-panel</code>	
Disable GPS support		<code>disable-gps</code>	
Disable Linux netlink socket capture (Prism2/ORiNOCO patched)		<code>disable-netlink</code>	
Disable Linux capture support		<code>disable-wireless</code>	
Disable libpcap capture support		<code>disable-pcap</code>	
Enable the system syspcap (not recommended)		<code>enable-syspcap</code>	
Disable suid-root installation		<code>disable-suid-root</code>	
Enable the use of WSP remote sensor		<code>enable-wsp100</code>	
Enable some extra stuff for Zaurus		<code>enable-zaurus</code>	
Force the use of local dumper code when ethereal is present		<code>enable-local-dumper</code>	
Support ethereal wiretap for logs (substitute the path to ethereal for DIR)		<code>with-ethereal=DIR</code>	
Disable support for ethereal wiretap		<code>without-ethereal</code>	
Enable support for the Advanced Configuration and Power Interface (ACPI)*		<code>enable-acpi</code>	

\* You must have Advanced Configuration and Power Interface (ACPI) enabled in Linux for the `enable-acpi` option to work.



In Linux, people sometimes refer to an option as a *switch*. The terminology you use is your choice, but the latter is more commonly used.

When you have finished configuring Kismet with the script, you are ready to do the following steps:

1. **If you haven't already done so, log in as root.**
2. **From the command prompt, run `make dep` to generate the dependencies.**
3. **Run `make` to compile Kismet using the GNU C Compiler (`gcc`).**
4. **Run `make install` to install Kismet.**

Now you are ready to use Kismet . . . almost. You still need to install the free GPSD. GPSD is the Global Positioning System Daemon, which provides spatial information from a GPS. This is useful for wardriving, especially after-the-fact. Without a GPS, you'll have all these discovered networks but you won't know how to find them again. GPSD is available for download from Russ Nelson at [www.pygps.org/gpsd/downloads](http://www.pygps.org/gpsd/downloads). The following steps show you how to install GPSD:

1. **Download `gpsd-1.10.tar.gz` or the latest version from Russ's Web site.**
2. **Make sure you are root.**  
Do a `su -` if you're not root.
3. **Type `gunzip gpsd-1.10.tar.gz` to uncompress the downloaded file.**
4. **Type `tar -xvf gpsd-1.10.tar` to untar the file.**
5. **Change to the directory you just created by typing `cd gpsd-1.10`.**
6. **Type `./configure` to execute the configure script.**
7. **Configure the GPSD binaries by typing `make`.**
8. **Copy the binaries to where you want by typing `make install`.**



You can make (no pun intended) sure that `gps` and `gpsd` are in the appropriate directories by issuing the `which gps` and `which gpsd` commands. The `which` output shows you the full path to the program so you can make sure you placed them appropriately.

9. **Turn off your computer and make sure your GPS is turned off, too.**
10. **Connect your GPS to your computer with the serial cable. (Of course, you can use a USB GPS as well.)**
11. **Turn on the GPS.**  
Give it time to acquire a signal.
12. **Reboot your computer.**

## 160 Part III: Advanced Wi-Fi Hacks



### 13. Start the GPS daemon by typing `gpsd -s 4800 -d localhost -r 2947 -p /dev/ttyS0`.

You need root privileges to start the GPS daemon.

This starts the daemon listening on port 2947. You can verify that it is running by port scanning, using the `netstat` or `ps` command, or typing `telnet localhost 2947`. Table 10-2 provides some `gpsd` command line options.



If you have a USB GPS, you should type `gpsd -p /dev/ttyUSB0`.

**Table 10-2** `gpsd` Command Line Options

<i>Option</i>	<i>Description</i>
-d	Debug level; you must specify a level.
-K	Keep-alive flag.
-p	Full path for the serial or USB GPS device.
-s	Baud rate. The most common rate is 9600, but you can specify different rates. But do so only when you know your GPS supports the rate.
-S	Port number where you want <code>gpsd</code> to open a listener. This is not the “listening port” for the <code>gps</code> itself, but for the GPS daemon or host.

### Configuring Kismet

Now are you ready to use Kismet? Well, not quite. You must first edit the Kismet configuration file, `kismet.conf`. Unlike other Linux programs, you need to configure Kismet before you use it. To configure Kismet, open and customize the `/usr/local/etc/kismet.conf` file using your favorite editor, for example, `vi`, `pico`, or `emacs`.



You need root privileges to edit the `kismet.conf` file.

You need to change at least the following options:

- ✓ **suiduser:** Look for the comment `# User to setid to (should be your normal user)`. As it says, type the name of a normal account, not root.
- ✓ **Support for your wireless card:** By default, Kismet is configured to support Cisco cards. If you don't have a Cisco card, you need to comment out the Cisco card support and then add your card. If you have an ORiNOCO or other Hermes chipset card, then uncomment the ORiNOCO line. Similarly, if you have a Prism2 card, then uncomment the Prism2 line.

- ✓ **Source:** By default, the ORiNOCO card uses `eth0` as the capture device, and the Prism2 card uses `wlan0`. If your system uses something else, like `eth1` or `eth2` for the ORiNOCO capture device or `wlan1` or `wlan2` for the Prism2, then you need to change the device in the configuration file. The format for this variable is *driver, device, description*. So, if you are one of those right-brained individuals, you might use `source=orinoco, eth1, AirPort`. If you have money to burn and can afford the best, then you might use `source=cisco, eth0, ciscosource`.
- ✓ **Channel-hopping interval:** By default, the channel-hopping interval (`channelvelocity`) is set to 5. This is the number of channels monitored every second. By increasing this value, you can monitor more channels per second. If you drive fairly fast, a lower value is better — with a high value, you will fly by and not get a good reading on the channel. If you want to monitor only one channel, then set `channelhop` to `false`, and it won't hop.
- ✓ **GPS support:** If you set up the GPS in the manner shown in this chapter, then the `gpshost` defaults are fine. Kismet is configured to use a serial device and listen on port 2947. If you don't want to use a GPS, then change the `gps` value to `false`.

If you want to change the sound Kismet plays when it finds a new access point, then change the `sound_alert` variable in the `kismet_ui.conf` file. Just type in the full path to the new `.wav` file you want to play. If you wish, you can change the user interface colors by altering the `kismet_ui.conf` file. You can use black, red, yellow, green, blue, magenta, cyan, or white, as well as bold (`hi-`). For instance, when you're worried about your battery status and you want the information to pop right out on the screen, change the `monitorcolor` variable to `hi-red` in the `kismet_ui.conf` file. This should make it more visible.



Should you wish, you can even use the Festival text-to-speech engine to report discovered networks. A little audio feedback is much safer when driving because you don't have to look at the laptop screen. You can find the Festival engine at [www.cstr.ed.ac.uk/projects/festival](http://www.cstr.ed.ac.uk/projects/festival).

### **Starting Kismet**

This time we're not kidding, you really are ready to use Kismet. Obviously, Kismet is harder to set up and use than NetStumbler (see Chapter 9). But you are now ready. You'd think that setting the `suid` to an account other than root that is the account that you'd use. Wrong. If you logged on with that account, then `su` to root.



If you try to use an account other than root to run Kismet, you don't have permission to set the PID number file (`kismet_server.pid`) in the `/var/run` directory. You need to gain root privileges. Most of the time, you use `su -`. However, do a `su` and not a `su -`. If you do the latter, you cannot write the

## 162 Part III: Advanced Wi-Fi Hacks

dump file. This is because when you use the latter, you change to root in the root environment. When you use the former, you gain root privileges, but maintain your normal user environment.

Now, to start Kismet, just type `kismet`. You can start Kismet with server (`kismet_server`) and client (`kismet_client`) options. If you don't know what options to use, type `kismet -help`. We have listed the options in Table 10-3 for your convenience.

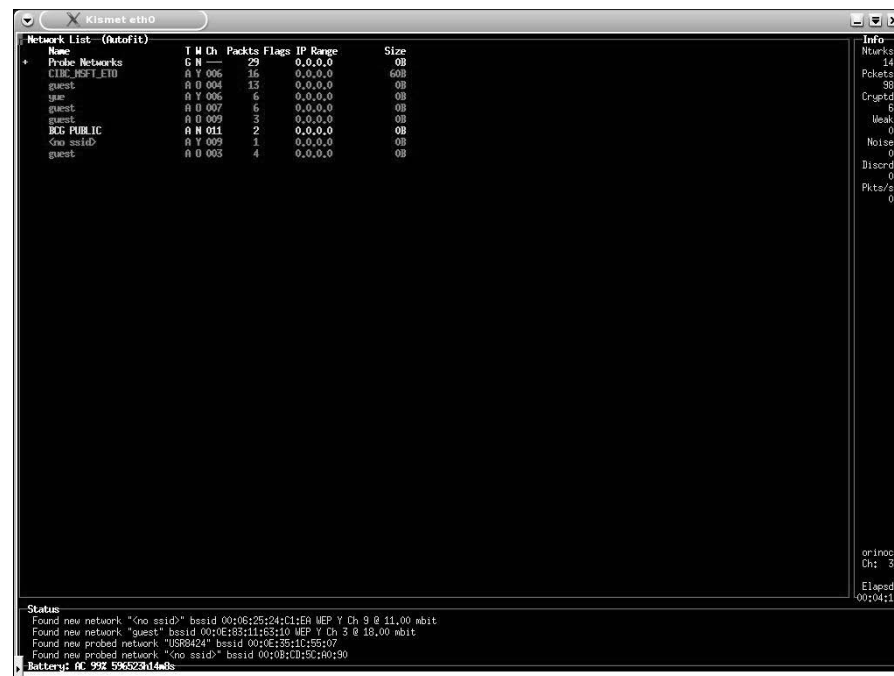
<i>Flag</i>	<i>Option Name</i>	<i>Description</i>
-a	allowed-hosts <hosts>	Comma-separated list of hosts allowed to connect
-c	capture-type <type>	Type of packet capture device; e.g., prism2, pcap
-d	dump-type <type>	Dumpfile type (wiretap)
-f	config-file <file>	Use alternative configuration file
-g	gps	GPS server; port or off
-h	help	The help file
-i	capture-interface <if>	Packet capture interface; e.g., eth0, eth1
-l	log-types <type>	Comma-separated list of types to log; e.g., dump, cisco, weak, network, gps
-m	max-packets <num>	Maximum number of packets before starting a new dump
-n	no-logging	No logging: process packets only
-p	port	TCP/IP server port for GUI connections
-q	quiet	Don't play sounds
-s	silent	Don't send any output to the console
-t	log-title <title>	Custom log file
-v	version	Kismet version

### Understanding the Kismet user interface

Figure 10-1 shows the information you get when Kismet is running. You can see that Kismet has three frames:

- ✓ Network List
- ✓ Info
- ✓ Status

These frames are described in the following paragraphs.



**Figure 10-1:**  
Kismet  
running.

### Network List frame

In the Network List frame you see the fields described in Table 10-4. In its help files, Kismet refers to this frame as the *Network display*. This frame takes up the majority of the Kismet user interface. If you are using a GPS, you see the GPS information in the bottom left-hand corner of this frame.

# 164

## Part III: Advanced Wi-Fi Hacks

<i>Field</i>	<i>Description</i>
Name	The BSSID or name of the wireless network sorted on the last time the network was seen
T	Type of WLAN detected: A = AP, H = Ad hoc, G = Group of wireless networks, D = Data only with no control packets, P = Probe request
W	WEP-enabled: Y = Yes, N = No
Ch	Channel of the device
Packets	Number of packets captured for the WLAN
Flags	Network attributes: A# = IP block found via ARP, U# = IP block found via UDP, D = IP block found via DHCP offer, C = Cisco equipment found, F = Vulnerable factory configuration
IP range	Self-explanatory
Size	Size of frame

If you see a ! (exclamation point) in front of a name, it's because Kismet saw activity in the last 3 seconds. If you see a . (period), that means Kismet saw activity within the last 6 seconds. Clearly, you are looking for those networks with an F flag!

Kismet starts in Autofit mode. In this mode, the names change automatically, and active ones appear first. However, you can sort the list by entering *s* at any time in the active window at the bottom. If you want to sort on SSID, enter *ss*. See Table 10-6 for the *s* command and others.



An interesting point on Kismet and encryption: Some access points don't accurately indicate the use of encryption by setting the correct bit in the 802.11 frame header. So, Kismet doesn't rely only on that bit. Instead, Kismet looks at the first few bytes of the logical link control (LLC) header to see whether they are the same. When they are, WEP is not used. When they are not, encryption is used.

### Info frame

In the Info frame on the right-hand side, you see the fields described in Table 10-5. In the help documentation, Kismet refers to this frame as the *Statistics frame*.

<b>Table 10-5</b>		<b>Info Fields</b>	
<i>Field</i>	<i>Description</i>		
Ntwrks	Total number of WLANs detected		
Pckets	Total number of packets captured		
Cryptd	Total number of encrypted packets captured		
Weak	Total number of packets with weak IVs (initialization vectors) captured		
Noise	Total number of garbled packets captured		
Discrd	Total number of packets discarded due to a bad CRC (ICV) value		
Pkts/s	Number of packets captured per second		
Ch:	Current channel		
Elapspd (Discon)	Time (hours:minutes:seconds) elapsed since the start of the capture		

You also see the type of card used for the capture. You will see `orinoc`, `prism`, or some other value.

#### ***Status frame***

The status frame lists the major events detected by Kismet. It is a scrolling display. For example, you will see messages as it finds networks.

You may also see the battery status. This feature is helpful when you are walking around with your laptop and not plugged in.

#### ***Commanding Kismet***

When you are running Kismet, you can use various commands to get more information. Table 10-6 provides the commands. If you're not sure of the commands, type `h` while Kismet is the active window.

<b>Table 10-6</b>		<b>Kismet Commands</b>	
<i>Command</i>	<i>Description</i>		
a	Show channel and encryption usage.		
c	Show information about wireless clients associated with an access point.		

(continued)

## 166 Part III: Advanced Wi-Fi Hacks

**Table 10-6 (continued)**

<i>Command</i>	<i>Description</i>
d	Print dumpable strings for quality, power, and noise.
e	List Kismet servers.
g	Group currently tagged networks together. Kismet prompts you for a new name.
h	Show the help window.
H	Return to normal channel hopping.
i	Get detailed information for a selected network.
l	Show wireless card power levels; that is, quality, power, and noise.
L	Stop channel hopping and stay on the current channel.
m	Mute sound.
n	Enter custom name.
Q	Quit.
r	Show the packet reception rate.
s	Sort network list.
t	Tag or untag current network or group.
u	Ungroup current group.
x	Close pop-up window.
z	Zoom network frame. This hides the info and status frames.

Kismet saves data automatically while it is running. When you are finished with Kismet, type Q to quit and close the application.

### ***Kismet logging***

By default, Kismet generates the following seven log files:

- ✓ **dump:** A raw packet dump.
- ✓ **network:** A plaintext log of detected networks.
- ✓ **csv:** A plaintext log of detected networks in comma-separated value format.

- ✓ **xml:** A plaintext log of detected networks in Extensible Markup Language (XML).
- ✓ **weak:** Weak packets detected and stored in AirSnort format. See Chapter 15 on what to do with this log.
- ✓ **cisco:** A log of Cisco equipment detected in Cisco Discovery Protocol (CDP) format.
- ✓ **gps:** A log of the GPS coordinates.

You can change logging by editing the `logtypes` variable in the `kismet.conf` file.

In Chapter 9, we discuss StumbVerter, which is an application that allows you to import NetStumbler's summary files into Microsoft's MapPoint 2004 maps. Well, StumbVerter doesn't work directly on Kismet logs. However, you can use a workaround: You can use WarGlue to convert your Kismet logs to NetStumbler format. From there you can export them to Summary format so StumbVerter can import them into MapPoint. You can get WarGlue at [www.lostboxen.net/warglue](http://www.lostboxen.net/warglue). It's developed by the Church of Wi-Fi (CoWF). Or download WarKizNiz from [www.personalwireless.org/tools/](http://www.personalwireless.org/tools/). WarKizNiz also accepts input from Kismet log files and converts them into NetStumbler .nsl format.

### *Shutting down Kismet*

Because Kismet requires your wireless card to be in monitor mode, you can't use the card to connect to a wireless network without either restarting PC Card services or rebooting your system. Alternatively, you can run `kismet_unmonitor` as root and then eject the card and reinsert it. This should reset its original network parameters. If not, then restart.



Remember, you must put your wireless card into RF monitor mode to use Kismet so your card cannot associate with a wireless network. If you need a network connection while running Kismet, then either use another wireless card to connect to a wireless network or an Ethernet card to connect to a wired network.

## *Installing and using Wellenreiter*

Wellenreiter is another wardriving tool. (A German-to-English dictionary translates *Wellenreiter* as *surfer*. Amazing how the subject of surfing keeps coming up in networking terminology.) Wellenreiter is a freeware user-friendly application, has a nice user interface, and is fairly straightforward to install and use.

## 168 Part III: Advanced Wi-Fi Hacks

Two versions of Wellenreiter are available:

- ✓ PERL scripts that support Linux. The script requires the `Net::Pcap` module (<http://search.cpan.org/~kcarnut/Net-Pcap-0.05/>) and the `GTK` or `GIMP Toolkit` ([www.gtk.org](http://www.gtk.org)). The latter is most likely already installed on most UNIX-based systems.
- ✓ C++ version that supports the wider UNIX world (C++/Qt) and handhelds such as Zaurus (C++/Opie).

The project is moving from PERL to C++.

Wellenreiter automatically handles the configuration and monitoring mode for most Cisco, ORiNOCO, and Prism2 cards. If you are using the PERL version, you can download, install, and run Wellenreiter by performing the following steps:

- 1. Download Wellenreiter-v1.9.tar.gz or the latest version from the Wellenreiter Web site ([www.wellenreiter.net](http://www.wellenreiter.net)).**
- 2. Make sure you are root.**  
If you're not, do a `su -`.
- 3. Uncompress the downloaded file by typing `gunzip Wellenreiter-v1.9.tar.gz`.**
- 4. Untar the file by typing `tar -xvf Wellenreiter-v1.9.tar`.**
- 5. Type `cd Wellenreiter-v1.9` to change to the directory you just created.**
- 6. Execute the Wellenreiter script by typing `perl wellenreiter.pl`.**

Surf on over to [www.wellenreiter.net/screenshots.html](http://www.wellenreiter.net/screenshots.html) to see some screenshots. Wellenreiter looks a lot like NetStumbler and Kismet, but we like its icons better. The left-hand pane lists the monitored channels, and the right-hand pane displays information about the wireless networks for the channels.

Wellenreiter saves a binary packet capture to the user's home directory. In our case — yours too, if you followed the steps above — the home directory is root, because we did the `su` to root. The binaries are in `pcap` format, which you can view with `ethereal` or `tcpdump`.

### *Using WarLinux*

WarLinux is a Linux distribution for wardrivers. It's available on diskette and bootable CD. It is recommend for system administrators who want to audit

## Wardriving, warwalking, and other war memes

The term *wardriving* was coined by Marius Milner as a play on the term *wardialing*. *Wardialing* in turn came from the 1983 movie *WarGames*, starring Matthew Broderick, Dabney Coleman, and Ally Sheedy. In the movie, Matthew, as nerdy David Lightman, unwittingly dials into a Department of Defense's war computer and almost starts a nuclear Armageddon. Forever after, hackers were portrayed as sitting at a computer connecting to networks.

Wardriving is also known as *NetStumbling* or *WiLDing* (*Wireless LAN Discovery*). (For more on WiLDing, see [www.bawug.org](http://www.bawug.org).)

But what is warwalking? Well, wardriving is the meme for other forms of network discovery. Warwalking is one of the mutations. *Warwalking* (<http://wiki.personaltelco.net/index.cgi/WarWalking>) is network discovery by walking around. No longer are the hackers sitting at their computers. They're out and about in your neighborhood.

Here are some other terms you might hear about:

- ✓ *Warcycling* ([www.maths.tcd.ie/~dwmalone/p/sageie-02.pdf](http://www.maths.tcd.ie/~dwmalone/p/sageie-02.pdf) /) is network discovery done from a motorcycle or bicycle.
- ✓ *Warflying* ([www4.tomshardware.com/column/20040430](http://www4.tomshardware.com/column/20040430)) is network discovery done from an airplane. (Because many of the antennae are omnidirectional, you

actually get some very interesting information from the air.)

- ✓ *Warkayaking*. There have even been reports (<http://wifinetnews.com/archives/003922.html>) of warkayaking around Lake Union in Seattle, Washington.
- ✓ *Warchalking* (<http://forbes.jiwire.com/warchalking-introduction.htm> or <http://webword.com/moving/warchalking.html>) is the marking of the pavement to denote the existence of an access point. This variant seems inspired by hoboes who, using shared pictographs during the Great Depression, would denote easy marks and the active presence of railroad detectives in chalk. Warchalking, however, is for wibos, not winos.
- ✓ *Warspying* ([www.securityfocus.com/news/7931](http://www.securityfocus.com/news/7931)) is when someone uses a X10 Wireless Technology receiver to capture the signals from wireless devices such as cameras. Makes you think twice about using those nanny-cams!

All you need to do is to think of a unique way to do network discovery to become famous. Hey, how about *warsurfing*? Not bad, but remember that water and electricity don't mix! (Oops. Actually, the term *warsurfing* [[www.netstumbler.org/showthread.php?t=2190](http://www.netstumbler.org/showthread.php?t=2190)] was used to indicate the practice of using Google to find NS1 files on the Internet.)

and evaluate their wireless network installations. The benefit of WarLinux is that you don't have to install Linux but can boot it from a diskette or CD-ROM.

You can find WarLinux at <https://sourceforge.net/projects/warlinux>.

## 170 Part III: Advanced Wi-Fi Hacks

### *Installing and using MiniStumbler*

NetStumbler, which we discuss in Chapter 9, was developed by Marius Milner. Well, Marius also developed MiniStumbler, a port for the Pocket PC. Marius calls NetStumbler and MiniStumbler “beggarware,” his term for free-ware. MiniStumbler is commonly used when warwalking (see the sidebar, “Wardriving, warwalking, and other war memes”) because it is versatile, user-friendly, and readily portable. It’s versatile because you can run other programs like CENiffer to grab packets and discover networks. It is user-friendly because it uses a Windows user interface and is fairly easy to install and use. It is portable because you can carry it places you could not take your laptop.

MiniStumbler offers the same functionality as its big brother NetStumbler. Like NetStumbler, MiniStumbler sends probe requests every second. This is known as *active scanning*, contrasted to Kismet’s passive scanning.

To use MiniStumbler, you need a handheld running HPC2000, Pocket PC 3.0, Pocket PC 2002, or Windows Mobile 2003. MiniStumbler supports Hermes (Avaya, Compaq, ORiNOCO, Proxim, TrueMobile) and Prism2 chipset cards, including Compact Flash. It also works with the built-in Wi-Fi of the Toshiba e740.

Due to its smaller screen size, MiniStumbler displays less information than its larger cousin. But the log files contain all the data that you get with NetStumbler. So, you can use MiniStumbler on your iPAQ to discover wireless networks, and use NetStumbler on your Windows XP laptop or desktop to view the data. Of course, you can dump the MiniStumbler output into StumbVerter (see Chapter 9) or use with Microsoft Streets & Trips.



MiniStumbler cannot read any of the text formats exported from NetStumbler.

Noticeably missing in MiniStumbler is MIDI support, so you cannot get that very gratifying audio feedback as it discovers access points.

You can find MiniStumbler at [www.netstumbler.com/downloads](http://www.netstumbler.com/downloads).

#### *Installing MiniStumbler*

To install MiniStumbler, copy the proper processor architecture from your host computer to the Pocket PC. Supported processor architectures include ARM, MIPS, and SH3. Typically, the install goes as follows:

- 1. Download MiniStumbler to the desktop.**
- 2. To install the program from the desktop, run `ministumbler installer.exe`.**

This action unpacks the files for the host computer and places the files in a special folder for the handheld.

**3. Select the directory where you want to install the program.**

Use the default unless you have a compelling reason not to do so.

**4. Click Yes to see the Readme file. If you'd prefer not to view the file, click No.****5. Click OK when you see the reminder message to check the handheld device.****6. Connect your handheld to the host computer.**

Activate your syncing software if it doesn't start automatically. The software should upload the Pocket PC files to the handheld.

**7. If the files are in the \*.CAB format, the Pocket PC where you uploaded the files will install the software.**

There is no setup routine; you're set to go.

That should do it for you. However, you may need to review your documentation for full installation instructions.

Obviously, you need a handheld that supports a wireless adapter, whether it is a PC Card or a Compact Flash card. Installing the drivers for the wireless adapter usually requires the use of the synchronization program as well. You probably don't want to buy multiple wireless cards (even though we have), so you might have an ORiNOCO card you already use with Kismet and NetStumbler. Regrettably using an ORiNOCO card is one of the difficult installations. You have to download the driver for the card (use version 7.x or greater) and unpack the files manually. Then you have to manually port the files to the handheld and install the \*.CAB file. Your device will warn you about unsafe drivers; just click OK. You did back up your Pocket PC, right? After you install the program and the drivers, you're set to go. Just locate the `ministumbler.exe` file under the Start or Start Programs menu and tap it.

***Configuring MiniStumbler***

After MiniStumbler executes, you should see the phrase `No AP` at the bottom of the window. This is good. Seeing `No wireless` is bad. (If you see the latter, make sure you have a working wireless card.) When MiniStumbler finds the first access point, you will see `1 AP`. If your GPS is working, you will also see the message `GPS on`.

After you start MiniStumbler, two words and three icons appear on the menu bar. The two words are

- ✔ **File:** This menu holds the file functions such as Open and Save. Also, the Enable Scan item is on this menu. Use it to enable or disable network scanning.
- ✔ **View:** This menu holds those options that affect how the interface looks; for example, toolbars, icons, and options. We cover each of these in detail later in this chapter.

## 172 Part III: Advanced Wi-Fi Hacks

The three icons are

- ✓ **Green arrow:** Enables or disables scanning.
- ✓ **Gears:** Automatically reconfigures the wireless card for scanning. It stops the Wireless Zero Configuration service and makes sure the card is set to a blank or “ANY” SSID.
- ✓ **Hand with Menu:** Opens the Options screen.

The Hand with Menu icon gives you access to the following tabs:

- ✓ **General:** Find options about how the scan is performed. Make sure you check the Reconfigure card automatically and Get AP Names options. These options have the same meaning as in NetStumbler. You also find options for scanning speed. The speed ranges from Slower (1) to Faster (5). The default is Medium (3). Adjust it to 1. As a general rule, use the following guidelines for adjusting the speed:
  - **Slower:** For warwalking.
  - **Slow:** For jogging or leisurely inline skating.
  - **Medium:** For inline skating or biking.
  - **Fast:** For low-speed driving up to 25 mph.
  - **Faster:** For driving over 25 mph.
- ✓ **Display:** This is a pull-down list box for display of GPS latitude and longitude. The default format is degrees and minutes to the one thousandth, that is, D°MM.MMM.
- ✓ **GPS:** In this tab, you find the information you need to make your GPS work with MiniStumbler. You can select the protocol, port (eight COM ports), baud, data bits, parity, stop bits, and flow control. The default is NMEA 0183, COM1, 4800 bps, 8 data bits, No parity bit, 1 stop bit.
- ✓ **Scripting:** You can write scripts to extend the functionality of MiniStumbler. Common scripting languages for Windows are VBScript and JScript.

If you have the correct configuration and a wireless device, MiniStumbler starts reporting as soon as you start the application.

### *Interpreting MiniStumbler*

If you’ve used NetStumbler, you’ll be comfortable with MiniStumbler. The user interface is essentially the same as the right-hand pane of NetStumbler. This means you have no filters to apply to the data; found in the left-hand pane of NetStumbler. MiniStumbler uses the same colors as NetStumbler. For example, detected networks show up as green, yellow, or red to indicate signal strength and gray for those out of range. The lock icon also means that the access point is WEP-enabled. Click the green arrow to pause discovery.

Unfortunately, MiniStumbler does not support the visualization tools that NetStumbler gives you. For example, you won't find the graph showing signal to noise over time.

Because of the small screen, you cannot see everything at once. You have to scroll to the right to see signal strength, SNR, and noise levels.

If you're set up to use a GPS, MiniStumbler also shows the latitude and longitude for all the wireless networks you find.

Selecting and right-clicking a Media Access Control (MAC) address opens a context menu that displays the SSID of the MAC. If you select an active MAC with an IP address or subnet assignment, then the menu provides three other Look up options regarding an IP address look-up at either ARIN, RIPE, or APNIC. Using this facility allows you to do a WHOIS query on the selected registration authority. It might help you determine whether you have a rogue access point.



You obviously need a network connection to the Internet to perform WHOIS queries on ARIN, RIPE, or APNIC.

When you exit MiniStumbler, it asks you whether to save the file. The file format is the same as that of NetStumbler: `YYYYMMDDHHMMSS.ns1`. Copy these files to NetStumbler and merge them with your other data.

## *Using other wardriving software*

We know some of you out there for one reason or another don't want to use NetStumbler, MiniStumbler, Wellenreiter, Kismet, or WarLinux. One good reason is that your organization has a policy against the use of freeware or open source software. That alone would preclude the use of those programs. Other wardriving tools are available to you, however. Some are free and some have a fee. Here is a sample:

- ✓ Aerosol ([www.sec33.com/sniph/aerosol.php](http://www.sec33.com/sniph/aerosol.php))
- ✓ AirMagnet ([www.airmagnet.com/products/index.htm](http://www.airmagnet.com/products/index.htm))
- ✓ AiroPeek ([www.wildpackets.com/products/airopeek](http://www.wildpackets.com/products/airopeek))
- ✓ Airscanner ([www.snapfiles.com/get/pocketpc/airscanner.html](http://www.snapfiles.com/get/pocketpc/airscanner.html))
- ✓ AP Scanner ([www.macupdate.com/info.php/id/5726](http://www.macupdate.com/info.php/id/5726))
- ✓ APsniff ([www.monolith81.de/mirrors/index.php?path=apsniff/](http://www.monolith81.de/mirrors/index.php?path=apsniff/))
- ✓ BSD-Airtools ([www.dachb0den.com/projects/bsd-airtools.html](http://www.dachb0den.com/projects/bsd-airtools.html))
- ✓ dstumbler ([www.dachb0den.com/projects/dstumbler.html](http://www.dachb0den.com/projects/dstumbler.html))
- ✓ gWireless (<http://gwifiapplet.sourceforge.net>)

## 174 Part III: Advanced Wi-Fi Hacks

---

- ✓ iStumbler (<http://istumbler.net>)
- ✓ KisMAC ([www.binaervarianz.de/projekte/programmieren/kismac](http://www.binaervarianz.de/projekte/programmieren/kismac))
- ✓ MacStumbler ([www.macstumbler.com](http://www.macstumbler.com))
- ✓ Mognet ([www.10t3k.net/tools/Wireless/Mognet-1.16.tar.gz](http://www.10t3k.net/tools/Wireless/Mognet-1.16.tar.gz))
- ✓ NetChaser ([www.bitsnbolts.com](http://www.bitsnbolts.com))
- ✓ Pocket Warrior ([www.pocketwarrior.org](http://www.pocketwarrior.org))
- ✓ pocketWinc ([www.cirond.com/pocketwinc.php](http://www.cirond.com/pocketwinc.php))
- ✓ Sniff-em ([www.sniff-em.com](http://www.sniff-em.com))
- ✓ Sniffer Wireless ([www.networkgeneral.com/](http://www.networkgeneral.com/))
- ✓ THC-Scan ([www.thc.org/releases.php?q=scan](http://www.thc.org/releases.php?q=scan))
- ✓ THC-Wardrive ([www.thc.org/releases.php?q=wardrive](http://www.thc.org/releases.php?q=wardrive))
- ✓ WiStumbler ([www.gongon.com/persons/iseki/wistumbler/index.html](http://www.gongon.com/persons/iseki/wistumbler/index.html))
- ✓ Wireless Security Auditor ([www.research.ibm.com/gsal/wsa](http://www.research.ibm.com/gsal/wsa))
- ✓ Wlandump ([www.guerrilla.net/gnet\\_linux\\_software.html](http://www.guerrilla.net/gnet_linux_software.html))

There is something for everyone on that list, regardless of whether you run Windows XP, Windows CE, SunOS, Red Hat Linux, FreeBSD, Mac OS, Zaurus, or Pocket PC.

## *Organization Wardriving Countermeasures*

In this and the previous chapters, you find out how to use several wardriving programs to discover wireless networks. You're probably thinking you can't do much to protect yourself against these applications. Well, you can, and that is what the remainder of the chapter will show you.

### *Using Kismet*

How can Kismet help with wardriving? Isn't it one of the problems? Don't people use it to discover my network? Well, yes, it is one of the problems. But look back at the description of Kismet. We said that you could use Kismet to find out whether others were running NetStumbler. Kismet sees and records the Probe Request. So here is your first countermeasure. Get yourself Kismet and look for others probing your wireless network. Commercial products like AiroPeek can help as well.

## *Disabling probe responses*

When a workstation starts, it listens for beacon messages to find an access point in range to send a beacon to. Even though the access point is sending about 10 beacons a second, this is not always enough to detect them because the workstation has to monitor 11 channels by going to each channel and waiting 0.1 seconds before moving to the next channel. Further, when your authenticated access point's signal starts to weaken, your workstation needs to find another access point. For this reason, the 802.11 standard authors created the Probe Request. Your workstation can send a Probe Request, which any access point in range will respond to with a Probe Response. The workstation quickly learns about all access points in range. Now imagine that you're not trying to associate, but you're just trying to find access points in range, and you understand how NetStumbler works. So, the countermeasure is quite obvious: Turn off Probe Response on your access point.

## *Increasing beacon broadcast intervals*

The beacon interval is a fixed field in the management frame. You can adjust the field to foil the fumbling stumblers. If someone drives by your access point and his or her device has to search all the channels and land on each for 0.1 seconds, then again the countermeasure is intuitive: Increase the beacon broadcast interval. This increases the likelihood that they won't grab your beacon when driving by.

## *Fake 'em out with a honeypot*

The term *honeypot* harkens back to childhood: Winnie the Pooh and his love for honey. Perhaps you remember how he found a pot of honey, put his head in, and got stuck. Imagine this same concept applied to your wireless network. You put an attractive system on the network to draw hackers like a Pooh-bear to honey. Invite the hackers in. While the hackers are exploring the system, you watch them and try to learn about them or their behavior. You can learn about honeypots at <http://project.honeynet.org>.



Human nature suggests you might want to strike back when you find someone attempting to breach your security. This is not a good idea. You cannot fight back and you might not want to anyway. Crackers often take over other sites so you may harm an innocent party. If you have evidence that someone is attempting to break in, contact the Secret Service, the FBI, or your local law enforcement agency.

It's easy to set up a honeypot system. Install some access point software on a computer and then create directories with names like `Payroll` or `*wars`.

## 176 Part III: Advanced Wi-Fi Hacks

### Turning the tables

As we often see, security tools are double-edged. Hackers have used Fake AP against hotspots. The hacker runs Fake AP on a laptop near a hotspot, say at a Starbucks. The clients wanting to use the Starbucks hotspot cannot discern the real access point from the cacophony of signals. This results in a denial of service to the hotspot's clients.

Don't turn on WEP and use a default SSID like `linksys`. A program like Fake AP ([www.blackalchemy.to/project/fakeap](http://www.blackalchemy.to/project/fakeap)) is useful for this purpose.

If one access point is good, then more is better. Black Alchemy developed Fake AP, which generates thousands of counterfeit 802.11b access points. Your real access point can hide in plain sight amongst the flood of fake beacon frames. As part of a honeypot or flying solo, Fake AP confuses NetStumblers and others. Because stumblers cannot easily determine the real access point, the theory is that they'll move on to the real low-hanging fruit — your neighbors. At least that is the theory. In real life, when you drive by a system with Fake AP, chances are it will not even register with NetStumbler. However, should you get stuck in traffic near the system, then that's a horse of a different color — you'll see the fake APs.

Fake AP runs on Linux and requires Perl 5.6 or later. You also need at least one Prism2 card with the CVS version of the Host AP Driver for Intersil Prism2/2.5/3 working. You can configure Fake AP to use dictionary lists for SSIDs and to generate WEP-encrypted and unencrypted access points.

If you're not Linux-inclined and prefer the Windows platform, you could use Honeyd-WIN32 ([www.securityprofiling.com/honeyd/honeyd.shtml](http://www.securityprofiling.com/honeyd/honeyd.shtml)), which creates fake access points and simulates multiple operating systems. And if you have some change burning a hole in your pocket, try KF Sensor ([www.keyfocus.net/kfsensor/](http://www.keyfocus.net/kfsensor/)).