

Wireless Tools

Until recently, network administrators mostly only had to worry about securing physical, fixed information technology assets. This includes servers, routers, and firewalls: the things that make up our wire-line networks. However, with the advent of inexpensive wireless network equipment, there is a whole new spectrum (no pun intended) of security problems to contend with.

This new technology has helped to lower the cost of deploying networks, brought access to places it wasn't before, and made the term "mobile computing" truly a reality. It has also drastically changed the network security perimeter for companies of all sizes. Traditionally, corporate networks were connected to the outside world in only a few places (see Figure 10.1). This allowed network managers to concentrate on protecting these limited access points. You could put firewalls and other defenses at these crucial choke points. The inside of the network was largely treated as trusted because there was no way to get there other than through the protected points.

Chapter Overview

Concepts you will learn:

- Wireless LAN terms
- The 802.11 protocols
- Weaknesses of wireless LANs
- Wireless assessment equipment

Tools you will use:

NetStumbler, StumbVerter, Kismet Wireless, and AirSnort

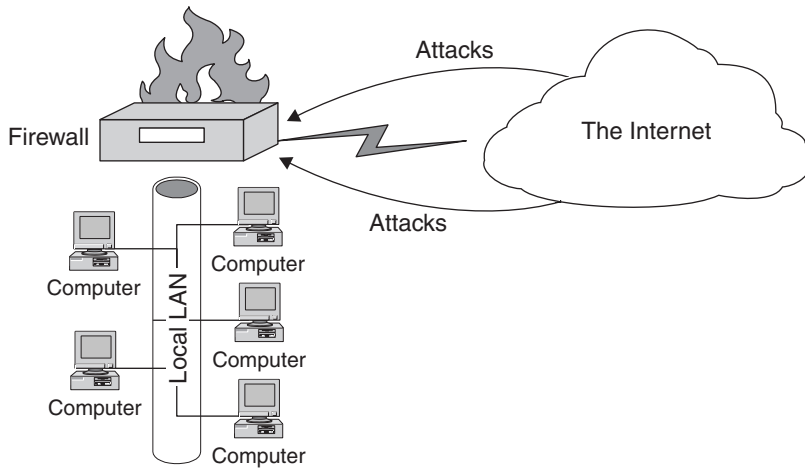


Figure 10.1 Network Threats Before Wireless Networking

Now the advancing march of technology has moved the security bar up a notch again. With a wireless LAN deployed, your new security perimeter becomes literally the air around you. Wireless attackers or eavesdroppers can come from any direction. If you have wireless access deployed, anyone with a \$50.00 card can potentially listen in on your network wire without ever stepping foot on your premises. Figure 10.2 shows the new network security perimeter with wireless technology. As you can see, if you are using wireless for part of your network, your security threats go up considerably. But before you can properly secure your wireless network, you need to understand how wireless local area networks function and what their basic weaknesses are.

Manufacturers of wireless LAN equipment have lowered the prices so much that it is now a feasible alternative for home networks. Rather than wiring your house for Ethernet to connect your PCs, you can buy a wireless base station and a couple of wireless cards and use the Internet from any room in your house (or outside for that matter). Many business conventions now offer free wireless Internet access to their attendees via wireless stations. There are grassroots campaigns to create free Internet access for neighborhoods outside the reach of DSL or cable by using public wireless access points. Wide deployment of wireless LAN technology is definitely here to stay, and sooner or later you will probably have to deal with it.

Wireless LAN Technology Overview

The most popular protocol for wireless LAN technology today is by far the 802.11 series, commonly known as **Wi-Fi**. The 802.11 wireless standards are basically an extension of the Ethernet protocol, which is why it interoperates so well with wired Ethernet networks. It uses the frequencies of 2.4GHz for 802.11b and 802.11g and 5GHz for 802.11a to

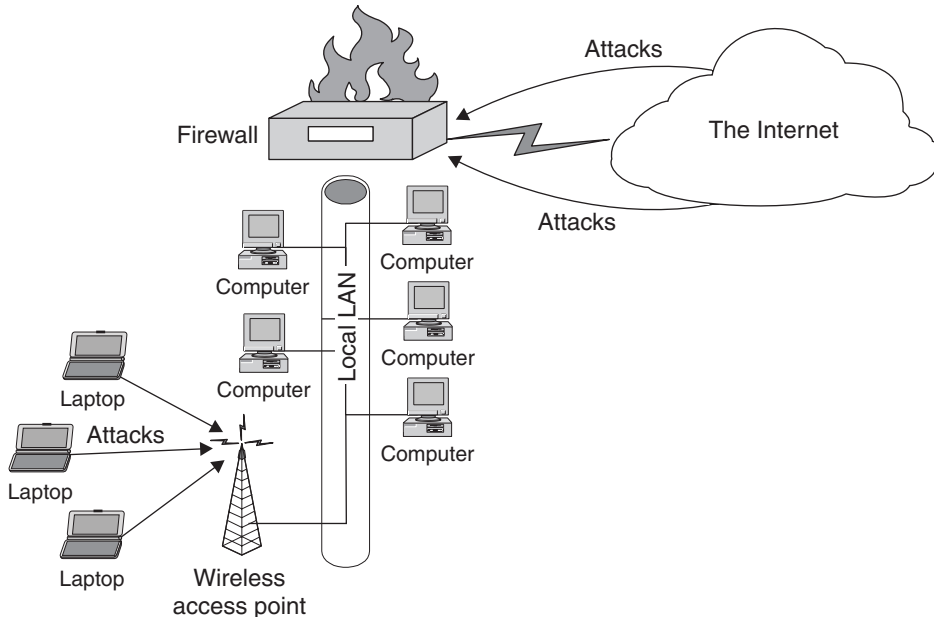


Figure 10.2 Network Threats with Wireless Networking

broadcast data signals. These frequencies are general-use spectrum, so you don't have to apply for a license from the FCC to use them. The downside of this is that other consumer devices can use these wavelengths too. Some cordless phones and microwaves are also on the 2.4GHz band, so if you have these devices or other Wi-Fi networks in your area, you may encounter some interference.

This wavelength is perfect for the short range that Wi-Fi is intended for. Its design parameters allow for about 150 feet indoors and over 800 feet outdoors under normal conditions. However, with a high-power antenna and line of sight, you can get up to a 20-mile range, which makes it attractive for office-to-office communications within a city (this assumes you are not in very mountainous terrain and you have access to a rooftop at least several floors up). Table 10.1 describes the four flavors of the 802.11 wireless standard that have emerged.

Wi-Fi Terms

A Wi-Fi wireless network can operate in one of two modes. **Ad-hoc mode** allows you to directly connect two nodes together. This is useful if you want to connect some PCs together and don't need access to a LAN or to the Internet. **Infrastructure mode** lets you set up a base station, known as an **access point** (AP), and connect it to your LAN. All of the wireless nodes connect to the LAN through this point. This is the most common configuration in corporate networks, as it allows the administrator to control wireless access at

Table 10.1 802.11 Wireless Standards

Standards	Descriptions
802.11a	This version of the standard uses the 5 GHz wavelength, which is a less crowded spectrum and is less likely to have interference problems. The theoretical potential for this technology is 54Mbps, which is a huge amount of bandwidth, but most applications in the field do not get that nearly that much.
802.11b	This is currently the most popular wireless standard. It uses the 2.4 GHz wavelength, which Bluetooth and other consumer devices also use. It offers up to 11Mbps of bandwidth, although practical applications under less than optimal conditions usually yield about half of that.
802.11g	A newer release, this standard provides up to 54Mbps bandwidth, but in the same 2.4GHz spectrum as 11b. It is also backwardly compatible with 11b hardware.
802.11i	This new protocol is basically an extension of 802.11b with fixes to the encryption protocol to make it much more secure. It has just recently been approved by the IEEE, and products using it should be available in late 2004.

one point. Each wireless access point and card has a number assigned to it called a **Basic Station System ID** (BSSID). This is the MAC address for the access point's wireless side. The access point also has a **Station Set Identifier** (SSID), which defines the name of the wireless network that all the nodes associate with. This name is not necessarily unique to that access point. In fact, most manufacturers assign a default SSID to APs so they are usable right out of the box. The access point's SSID is needed to connect to the network. Some base stations have additional functionality, including routers and built-in DHCP servers. There are even some integrated units that act as a wireless access point, firewall, and router for home and small business users.

You set up a wireless network node by installing a wireless **network interface card** (NIC) in a computer. A wireless NIC comes in several forms: It can be a card that goes in a PC slot, a PCMCIA card, an external USB device, and now even a compact flash format for the smaller slots in handheld computers. An 802.11 wireless network in infrastructure mode has an access point that acts as your bridge between the wired Ethernet LAN and one or more wireless endpoints. The access point sends out "beacon" broadcasts frequently to let any wireless node in the area know that it is there. The beacon broadcasts act like a lighthouse, inviting any wireless nodes in the area to log on. These beacon signals are part of the problem with Wi-Fi. It is impossible turn off these signals completely, which makes it hard to hide the fact that you have a wireless network in your office. Anyone with a wireless card can at least see your beacon signals if they are in range, although some sets allow you to limit the amount of information that goes out in these broadcasts.

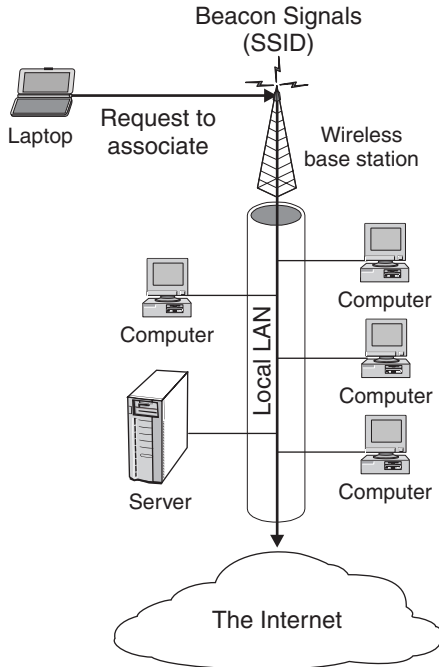


Figure 10.3 Wireless Network Operation

These signals contain basic information about the wireless access point, usually including its SSID (see Figure 10.3). If the network isn't using any encryption or other protections, then this is all that is required for someone to access to the network. However, even on an encrypted wireless network, the SSID is often transmitted in the clear and the encrypted packets may still be sniffed out of the air and subject to cracking attempts.

Dangers of Wireless LANs

While they offer flexibility and functionality that a wired LAN can't offer, they also introduce some unique challenges and dangers to the security-minded network administrator. Here are some things to consider when adding wireless LANs to your infrastructure.

Eavesdropping

The easiest thing for a hacker to do to a wireless network is to gather packets using a wireless sniffer. There is very little you can do about this, barring encircling your building in lead shielding! The designers of wireless networks did think about this, and built into the design an encryption standard called **Wired Equivalent Privacy** (WEP) so that the data could be encrypted. Unfortunately, a fundamental flaw in the way the algorithm works

makes it potentially crackable (one of the tools later in this chapter demonstrates this). So even with WEP running, any data that travels over a wireless network is potentially subject to inspection by outsiders. Someone could listen over your wireless link, sniffing for log-ins, passwords, or any other data.

Access to Wireless PCs

A wireless link gives potential attackers a vector into a machine on your network. Besides the access points, machines with wireless cards can sometimes be seen from the outside. Using this mode of access, they can launch attacks against a machine that is probably not protected by your firewall and may not be locked down like your perimeter defenses or public servers.

Access to the LAN

This is probably the biggest danger that wireless networks present. If hackers can get access to your LAN via a wireless access point, they often have the keys to your kingdom. Most LANs run an unrestricted DHCP server, so hackers can get a valid IP address and begin exploring your network. They can then run vulnerability scanners or port scanners such as Nessus and Nmap to find machines of interest and to find holes to exploit.

Anonymous Internet Access

Even if hackers are not interested in what is on your LAN, they can use your bandwidth for other nefarious uses. By logging onto your network and then accessing the Internet, they can hack and do whatever damage they wish to do without it being traceable back to them. Any attacks or mischief perpetrated from this connection will be traced to your network. The authorities will come knocking on *your* door, not theirs. This method of hacking will become more common as hackers realize how hard it is to trace attacks originating in this manner. There is little chance of catching someone coming from a wireless network unless you have expensive triangulation equipment in place beforehand. Unsecured wireless LANs offer hackers the best anonymous access there is.

802.11-Specific Vulnerabilities

In addition to the basic insecurities of wireless LANs, there are some problems specific to the 802.11 standard. Some of these are due to the manufacturer's bad design or default configurations. Other issues are due to problems with the standard's overall design.

Default SSIDs Each Wi-Fi base station has a specific identifier that you must know to log onto the network. This provides some level of security if it is implemented properly. Unfortunately, many people fail to change the default SSID set by the manufacturer. It is easy to find networks with the manufacturer's default SSID, such as `linksys`, `default`,

and so on. When hackers see this, they can assume that the administrator didn’t spend much time setting up and securing the wireless network.

Beacon Broadcast Beacon broadcasts are an inherent problem with wireless networks. The base station must regularly broadcast its existence so end user radios can find and negotiate a session, and because the legitimate user devices have not been authenticated yet, this signal must be broadcast in the clear. This signal can be captured by anyone, and at a minimum they then know that you have a wireless LAN. Many models let you turn off the SSID portion of this broadcast to at least make it a little harder for wireless eavesdroppers, but the SSID is still sent when a station is connecting, so there is nonetheless a small window of vulnerability.

Unencrypted Communications by Default Most wireless LAN devices today offer the option of turning on the built-in wireless encryption standard WEP. The problem is this usually has to be turned on manually. Most manufacturers ship their equipment with it off by default. Many administrators are in a hurry to set up a wireless network and don’t take the time to enable this important feature. If a nontechnical person is setting up the network, the chances are almost nil that the encryption will get turned on. There is also the issue of sharing the secret key with all your users, since WEP uses a single key among all users. This can be an administrative nightmare if you have a lot of users connecting wirelessly.

Weaknesses of WEP Even when the built-in encryption is used, the signal is still at risk of being read. There are some fundamental weaknesses in the implementation of the encryption algorithm in WEP that allows it to be broken after a certain amount of traffic is intercepted. These weaknesses have to do with the way the keys are scheduled. WEP uses weak initialization vectors (IVs) at a high enough rate that it eventually becomes possible to crack the key. Once the encryption is broken, not only can attackers read all the traffic traversing the wireless network, they can probably log on to the network. So while WEP offers some basic protection against casual eavesdroppers, any serious interloper is going to have software to potentially crack the encryption.

The “War-Driving” Phenomenon

Searching for unsecured wireless LANs has become a popular pastime among hackers and wireless hobbyists. This practice, akin to earlier hackers mass dialing or **war dialing** random banks of telephone numbers to find active modems, has become known as **war driving**. Mostly what wireless hackers do is drive around with a wireless card and some software waiting to pick up a signal from a network. The software can log the exact location of the wireless network via GPS, as well as lots of other information such as if it is encrypted or not. If the wireless LAN doesn’t have encryption or other protections turned on, war drivers can surf the Internet or explore the local LAN over the wireless link. There is not a high skill level required to do this, so it appeals to all levels of the hacker ranks.

Companies using wireless LANs in dense environments around their offices or near major roads and freeways are at the most risk from this kind of activity. This would include offices in urban environments and downtown areas where there are a lot of high rises. Wireless networks using 802.11b have an effective distance of a couple hundred yards. This can easily bridge the space between two buildings or several floors in a high rise. In a crowded downtown area, it is not uncommon to find several unprotected wireless LANs inside a building. From a security standpoint, tall buildings tend to be one of the worst places to run a wireless LAN. The typical glass-windowed building allows the signals from your LAN to travel quite a distance. If other buildings are nearby, it is almost a sure thing that they will be able to pick up some of your signals. Even worse are tall buildings around a residential area. Imagine teenagers and other ne'er-do-wells scanning for available wireless LANs from the comfort of their bedrooms in suburbia.

A recent study found that over 60% of wireless LANs are completely unsecured. War drivers have even taken to posting the wireless access points they find to online databases with maps so anyone can find open wireless LANs just about anywhere in the country. They categorize them by equipment type, encrypted or not, and so forth. If you have a wireless LAN in a major metropolitan area, it's a good chance that it is cataloged in a system like this, just waiting for an opportunistic hacker in your area with some time on his hands. The following are some of the online databases you can check to see if your company's wireless LANs are already cataloged.

- www.shmoo.com/gawd/
- www.netstumbler.com/nation.php

Note that most sites will remove your company's name from the listing if you request it.

Performing a Wireless Network Security Assessment

It would be easy for me to tell you that due to the security dangers of wireless networking, you should just not allow any wireless access on your network. However, that would be analogous to telling you to stick your head in the sand and hope the problem will go away. Wireless access is not going away. It is one of the hottest areas for growth and investment in the technology area. Vendors are churning out wireless adapters for all kinds of devices at a scary and ever-cheaper rate. Many retail companies such as McDonald's and Starbucks are installing wireless access points in their stores to attract customers. Intel Centrino laptops have a wireless radio built right in. Your users will come to expect the freedom that wireless LAN technology brings. They will want to be able to log on with their wireless-enabled laptops anytime, anywhere. This means that you are going to have to deal with your wireless security sooner or later. The tools in this chapter will help you assess your wireless network security and take steps to improve it if need be. It will also help you to deploy a wireless LAN solution more securely if you are doing it for the first time.

Equipment Selection

To perform wireless network security assessments, you will need at a minimum a wireless network card, a machine to run it on, and some software.

Wireless Cards Most of the software covered in this chapter is free, but you will have to buy at least one wireless network card. There are many different manufacturers to choose from and prices are quite competitive. Expect to pay from \$40 to \$80 for a basic card. You will want to carefully research your choice of manufacturers and models because not all cards work with all wireless software packages.

There are basically three different chipsets for 802.11b devices. The Prism II chipset by Intersil is probably the most common and is used by Linksys, the largest manufacturer of consumer wireless cards. The Lucent Hermes chipset is used in the WaveLAN and ORiNOCO cards and tends to be in higher-end corporate equipment. Cisco has its own proprietary chip, which has some special security features. The Prism II cards will work on Kismet wireless, the Linux software reviewed in this chapter, but not on the Windows platform. D-Link cards work with Windows but not with the Windows security toolkits that are commonly available. Also, models of particular manufacturers can be important. The older Linksys USB cards used a different chipset and do not work on well Linux.

To add to this confusion, some of the newer protocols aren't supported yet by many packages. The current versions of the software packages reviewed in this chapter don't support the newer 802.11g standard. The major vendors have yet to release their interface code for software developers to write to. Once they do, the drivers should become available shortly thereafter. You should check the respective software Web sites before purchasing your equipment for supported cards and protocols. For purposes of these reviews, I used the ORiNOCO Gold PCMCIA card, which works well with both the Windows and Linux software.

Hardware and Software In terms of hardware to load the software on, just about any decently powered machine will do. The UNIX software ran fine for me on a PII 300 with 64MB of ram. The Windows software should also run on a system like this. You should definitely load the software on a laptop since you are going to be mobile with it. There is a Palm OS version of Kismet Wireless and a Pocket PC version of NetStumbler available, so you can even put them on palmtops. There are now wireless cards available for both major platforms (Palm and Pocket PC) of the smaller handheld computers that can take advantage of this software.

You should also make sure you have plenty of hard disk space available if you intend to attempt cracking WEP keys. This requires anywhere from 500MB to several gigabytes of space. Be careful not to leave the machine unattended if you are sniffing wireless data and don't have a lot of extra space—you could easily fill up your hard drive and crash the computer.

If you are auditing your wireless perimeter and want to know exact locations, you may also consider getting a small handheld GPS receiver. Make sure your GPS device has

an NMEA-compatible serial cable to interface with your laptop. With this hardware, you can log the exact points from which your wireless access points are available. The products covered in this chapter have the capability to take GPS data directly from the receivers and integrate it into the output. Finally, if you can spring for GPS-compatible mapping software such as Microsoft MapPoint, you can draw some really nice maps of your assessment activity.

Antennas For wireless sniffing around the office, the built-in antennas on most cards work just fine. However, if you really want to test your wireless vulnerability outdoors, you will want an external antenna that lets you test the extreme range of your wireless network. After all, the bad guys can fashion homemade long-range antennas with a Pringles can and some PVC. You can buy inexpensive professional-grade wireless antennas from several outfits. I bought a bundle that came with the ORiNOCO card and an external antenna suitable for mounting on the top of a car.

This is another reason you need to choose your wireless card carefully. Some cards allow external antennas to be attached but others do not. You should be sure the card(s) you purchase have a port for one if you intend to do wireless assessments. Cards known to allow external antennas are the ORiNOCO mentioned earlier as well as the Cisco, Samsung, and Proxim cards.

Now that you have the background and the gear, let's check out some free software that will let you get out there and do some wireless assessments (on your own network, of course!).



NetStumbler: A Wireless Network Discovery Program for Windows

NetStumbler

Author/primary contact: Marius Milner

Web site: www.netstumbler.org/

Platform: Windows

License: Freeware

Version reviewed: 0.3.30z

NetStumbler forums: <http://forums.netstumbler.com/>

NetStumbler is probably the most popular tool used for wireless assessments, mainly because it is free and it works on the Windows platform. In fact, it is so popular that its name has become synonymous with war driving, as in "I went out NetStumbling last night." I guess the author so-named it because he "accidentally" stumbled on wireless networks while using it.

NetStumbler isn't considered truly open source since the author doesn't currently make the source available. However, it is freeware and it is worth mentioning since it's the most widely used tool on the Windows platform. There are many open source add-ons

available for it (one of these is discussed later in this chapter). It also has a very open source mentality in terms of its user community and Web site. The Web site is highly informative and has lots of good resources for wireless security beyond just the program. There is also a mapping database where other NetStumblers enter access points that they found while using the program. If your company's wireless network is in the database and you want it removed, they will be happy to do that for you.

Installing NetStumbler

1. Before installing NetStumbler, make sure you have the correct drivers installed for your wireless card. On newer versions of Windows, such as 2000 and XP, this is usually pretty straightforward. Install the software that came with your card and the system should automatically recognize the card and let you configure it. Support for Windows 95 and 98 can be dicey. Check your card's documentation for specifics.
2. Once your card is up and working, verify it by attempting to access the Internet through a wireless access point. If you can see the outside world, then you are ready to start installing NetStumbler.
3. The NetStumbler installation process is as easy as installing any Windows program. Download the file from the book's CD-ROM or www.netstumbler.org and unzip it into its own directory.
4. Execute the setup file in its directory and the normal Windows installation process begins.



When the installation is complete, you are ready to start Netstumbling.

Using NetStumbler

When you start NetStumbler, the main screen displays (see Figure 10.4).

In the MAC column, you can see a list of access points NetStumbler has detected. The network icons to the left of the MAC address are lit up green if they are currently in range. The icon turns yellow and then red as you pass out of range. Inactive network icons are gray. The graphic also shows a little lock in the circle if that network is encrypted. This gives you a quick way to see which networks are using WEP. NetStumbler gathers additional data on any point that it detects. Table 10.2 lists the data fields it displays and what they signify.

As you go about your network auditing, the main NetStumbler screen fills up with the wireless networks that you find. You will probably be surprised at the number of networks that show up around your office. And you will be even more surprised at how many have encryption turned off and are using default SSIDs.

The left side of the screen displays the different networks detected. You can organize them using different filters. You can view them by channel, SSID, and several other criteria. You can set up filters to show only those with encryption on or off, those that are

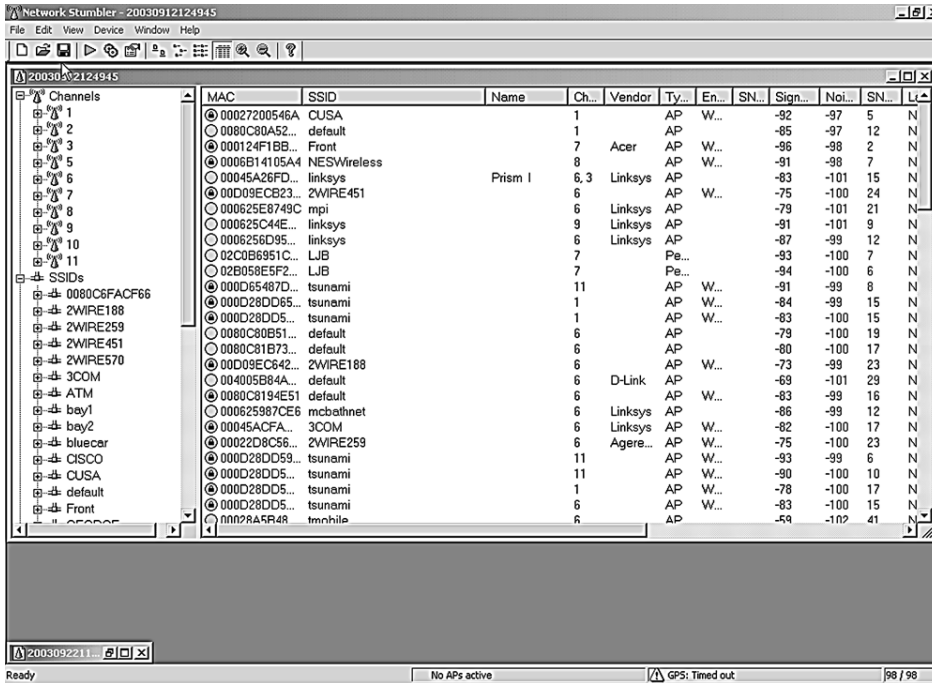


Figure 10.4 NetStumbler Main Screen

Table 10.2 NetStumbler Data Fields

Data Fields	Descriptions
MAC	The BSSID or MAC address of the base station. This is a unique identifier assigned by the manufacturer, and it comes in handy when you have a lot of stations with the same manufacturer default SSID such as linksys.
SSID	The Station Set Identifier that each access point is set up with. This defines each wireless network. You need this to log on to any wireless network, and NetStumbler gladly gathers it for you from the beacon signal. As noted in the MAC field description, this is not necessarily a unique ID since other base stations may have the same SSID. This could be a problem if two companies in the same building are using default SSIDs. Employees may end up using another company's network or Internet connection if it is not set up correctly with a unique SSID.
Name	The descriptive name, if any, on the access point. Sometimes the manufacturer fills this in. The network owner can also edit it; for example, Acme Corp Wireless Network. Leaving this name blank might be a good idea if you don't want people knowing your access point belongs to you when they are war driving around.

Data Fields	Descriptions
Channel	The channel the base station is operating on. If you are having interference problems, changing this setting on your access point might eliminate them. Most of the manufacturers use a default channel. For example, Linksys APs default to 6.
Vendor	NetStumbler tries to identify the manufacturer and model of the wireless equipment found using the BSSID.
Type	This tells you whether you found an access point, a network node, or some other type of device. Generally you will be finding access points that are signified by AP. Wireless nodes show up on here as Peer. This is why, even without a wireless network set up, having wireless cards in your PC can be risky. Many laptops now come with built-in wireless radios, so you may want to disable these before they are initially deployed if the users are not going to be using them.
Encryption	This shows what kind of encryption the network is running, if any. This is very important; if the network isn't encrypted, outsiders can pull your network traffic right out of the air and read it. They can also log onto your network if other protections aren't in place.
SNR	Signal-to-Noise ratio. This tells you how much other interference and noise is present at the input of the wireless card's receiver.
Signal	The signal power level at the input to the receiver.
Noise	The noise power level at the input to the receiver.
Latitude	Exact latitude coordinates if you are using a GPS receiver with NetStumbler.
Longitude	Exact longitude coordinates if you are using a GPS receiver with NetStumbler.
First seen	The time, based on your system clock, when the network's beacon was first sensed.
Last seen	NetStumbler updates this each time you enter an access point's zone of reception.
Beacon	How often the beacon signal is going out, in milliseconds.

access points or peers (in ad-hoc mode), those that are CF pollable (provide additional information when requested), and any that are using default SSIDs.

On the bar along the bottom of the main screen you can see the status of your wireless network card. If it is functioning properly, you will see the icon blinking every second or so and how many active access points you can see at that moment. If there is a problem with the interface between your network card and the software, you will see it here. On the far right of the bottom bar is your GPS location if you are using a GPS device.

The blinking indicates how often you are polling for access points. NetStumbler is an active network-scanning tool, so it is constantly sending out “Hello” packets to see if any wireless networks will answer. Other wireless tools, such as the Kismet tool discussed later in this chapter, are passive tools in that they only listen for the beacon signals. The downside of the active tools is that they can miss some access points that are configured not to answer polls. The upside of an active scanning tool is that some access points send out beacon signals so infrequently on their own that you would never see them with a passive tool. Also, keep in mind that active polling can set off wireless intrusion detection systems. However, very few organizations run wireless detection systems, and if you are using NetStumbler only as an assessment tool for your own network, then being stealthy shouldn’t be that important to you.

If you click on an individual network in this mode it shows a graph of the signal-to-noise ratios over the times that you saw the network. This lets you see how strong the signal is in different areas (see Figure 10.5).

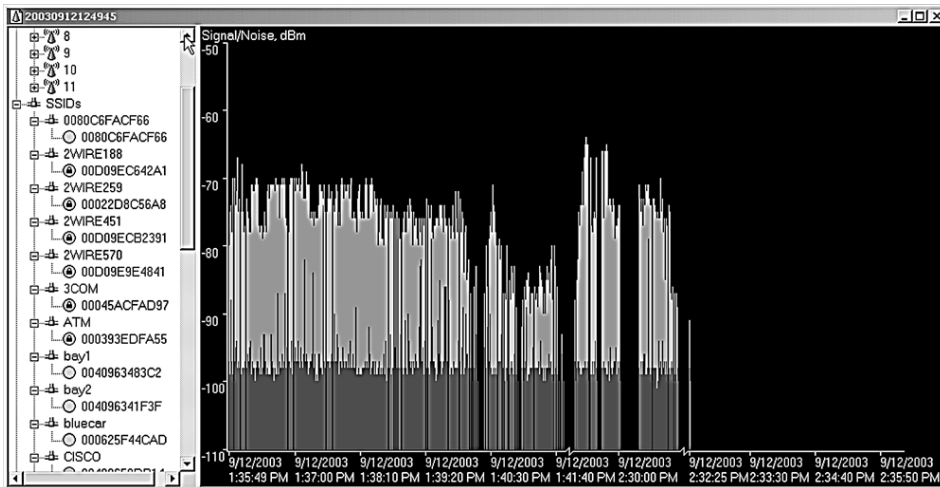


Figure 10.5 NetStumbler Signal Graph

NetStumbler Options

Under the View menu, select the Options submenu to display the dialog box for setting NetStumbler options. Table 10.3 lists the tabs and the choices available.

Tips for Effective—and Ethical—Wireless Auditing

Get Permission

Make sure you have permission from management to do your wireless assessment. If you are an outside consultant, you should have a letter of permission or engagement signed by upper management. If the company does not own the building, get management to clear it with building security so you have permission to be on the premises.

Determine Your Wireless Perimeter

Walk the entire perimeter and find out how far your signal goes. (A good rule of thumb is to go only in publicly accessible places that wireless crackers or war drivers would have access to.) If possible, get a map and mark your wireless perimeter on it.

Table 10.3 NetStumbler Options

Tabs	Descriptions
General	Set the rate of polling for your access points. You can also set it to auto-adjust based on your speed if using GPS. There is an option to automatically reconfigure your card when a new network is found, but you probably don't want to do this in a busy area—if there are a lot of access points around, your card will be changing configuration every few seconds and it will slow your computer down. Also, the software may end up configuring your card for a foreign network and you could be trespassing inadvertently. Not cool! (See the sidebar on "Tips for Effective—and Ethical—Wireless Auditing".)
GPS	Set up your GPS receiver to interface with NetStumbler. I used a Meridian handheld GPS with a serial cable. All I had to do was set the right port and communication settings and NetStumbler started importing the data right away.
Scripting	Set up to call external scripts. You can use Visual Basic or any number of Windows-based languages to do additional things based on the NetStumbler output. External programs can also use this functionality.
MIDI	You can configure NetStumbler to play the signal-to-noise ratio as a Midi file. I'm not sure why you'd want to do this as it could get noisy in an area with a lot of networks, but I guess you could use it to home in on a elusive signal by sound.

Start outside what you think is a reasonable reception range and work your way in. Make a broad circle around your business premises and work your way in to find out how far out the signal goes. Then go back and make a broader circle to see if any pockets of reception extend out farther.

Sometimes quirks in the landscape or manufactured objects can cause weird extensions of the signal: it can be reflected or focused by buildings, billboards, trees, and other objects. Assume the war drivers take advantage of this.

Once you've established the perimeter, you can evaluate the pockets of reception and take steps to eliminate or reduce them. Sometimes you can decrease the distance the signal goes by moving your access points to an interior room or to the other side of the building. As mentioned earlier, many units let you adjust the signal strength to limit radiation from the building.



Flamey the Tech Tip:

Be a Good Wireless Network Neighbor

When auditing your own network, it is likely that you will come across other wireless access points and nodes in the nearby area or building. Some of them will be unsecured.

Be a good neighbor and let them know that they have an unsecured access point. They may not even be aware of the dangers this poses.

Be a good neighbor and don't attempt to surf their network to demonstrate how bad their security is. Not only is this very bad behavior, but it could get you put in jail if you are caught. So resist the temptation and be a good wireless network neighbor.

Use an External Antenna

Using a card that supports the addition of an external antenna extends your range dramatically. These cards don't cost much more than the cheapest wireless NICs. The consumer varieties, such as Linksys or D-Link, generally don't support this, but it is worth paying an extra \$100.00 for a better card. If you are really strapped, there are Web sites that tell how to make a homemade antenna for your card. Assume that your opponents will be able to find these sites too and will have at least as good an antenna as yours.

Audit Under Optimal Conditions

Rain, humidity, and smog can affect wireless transmission. The wavelength that 802.11b operates on resonates in water, and that can dull a signal in a rainstorm or even when there is a lot of moisture in the air. Tree leaves, due to their high water content, have the same effect. Your results in the winter may be different from those in the summer. Pick a clear, dry day to test to optimize your results.

Saving NetStumbler Sessions

NetStumbler automatically starts saving your session each time you open it. This lets you examine your NetStumbler sessions at another time. By default, sessions are saved in a native NetStumbler format. You can also save the sessions as text for importing into a spreadsheet or word processor and in the wi-scan format, which is a budding file standard for wireless sniffing logs. You can also export them in a number of formats.

NetStumbler assigns a unique number that is a combination of the date and time for each session at the top of the window (see Figure 10.5). This is helpful for tracking your sessions and results. You can change this name to something more descriptive if you like.

Now that you have a lot of data about your wireless perimeter, you may want to produce some reports, either for management or for a customer if you are doing this as a consultant. If you have been collecting GPS data, you can create some nice maps with the Microsoft MapPoint program and the open source tool discussed next.



StumbVerter: A Map Conversion Program for NetStumbler

StumbVerter

Author/primary contact: Michael Puchol; Sonic Security

Web site: www.sonar-security.com/

Platform: Windows

License: Freeware (GPL-like)

Version reviewed: 1.5

Mailing list:

Send a blank e-mail to stumbverter-subscribe@c2security.org.

StumbVerter is a neat little program that takes the output from NetStumbler and converts it into input for the Microsoft MapPoint program. It has functionality beyond the basic NetStumbler program, including:

- Access points shown as little beacons on the map.
- Beacons displayed in various sizes and colors depending on the APs strength and WEP mode.
- Balloons for logging notes and other information.
- Navigational information such as speed, heading, and distance to the nearest known AP.
- An antenna comparison tool.

You must have a legal license for Microsoft MapPoint 2002 software to use StumbVerter. I know this is getting away from the idea of free software, but the functionality this

adds is well worth the extra \$200.00 that MapPoint will set you back. And of course, the StumbVerter software itself is freeware. Several projects are underway to develop a program to convert NetStumbler files into something free, such as a MapQuest or MapBlast map (but none of these were far enough along as of publication to include). At any rate, if you have to present reports to management, the color maps will definitely help your case.

Installing StumbVerter

1. Make sure you have Microsoft MapPoint and NetStumbler installed before attempting to install StumbVerter. It will not load correctly without these two programs. If you just installed these, reboot your computer.
2. You must also be operating with a GPS receiver and logging that information into NetStumbler. In order for StumbVerter to be able to do anything the data, it must have the GPS coordinates of the wireless networks. This is how it figures out where to put the graphics.
3. Download StumbVerter from the book's CD-ROM or the Web site and unzip it.
4. Double-click on the setup file and it will install it on your system.



Once you have all these installed, you can start working with NetStumbler and StumbVerter.

Using StumbVerter

1. To use StumbVerter, you need some data to map. So go out with NetStumbler and collect some data on your wireless networks.
2. Save the session in NetStumbler and export it in text summary format.
3. Start StumbVerter by double-clicking its icon on your desktop.
4. On the menu at the top of the screen, click on Map, select Create New, then pick your region.
5. Once the map loads, click on Import and select the .nsi file that represents the NetStumbler session you want to map. StumbVerter displays the logged data graphically as a map (see Figure 10.6)

Green towers represent encrypted access points; red towers represent unencrypted access points. The signal strength is shown by the waves coming out of the top of the icon: the more waves, the stronger the signal.

If you single-click on a specific access point, the map centers on that point and shows you the informational balloon. Initially, this shows the network's SSID. Double-clicking on it shows all the notes associated with that AP and lets you add comments.

The View menu has several options for manipulating and cleaning up your map. For example, you can remove the Points Of Interest (POIs) that MapPoint inserts, unless you

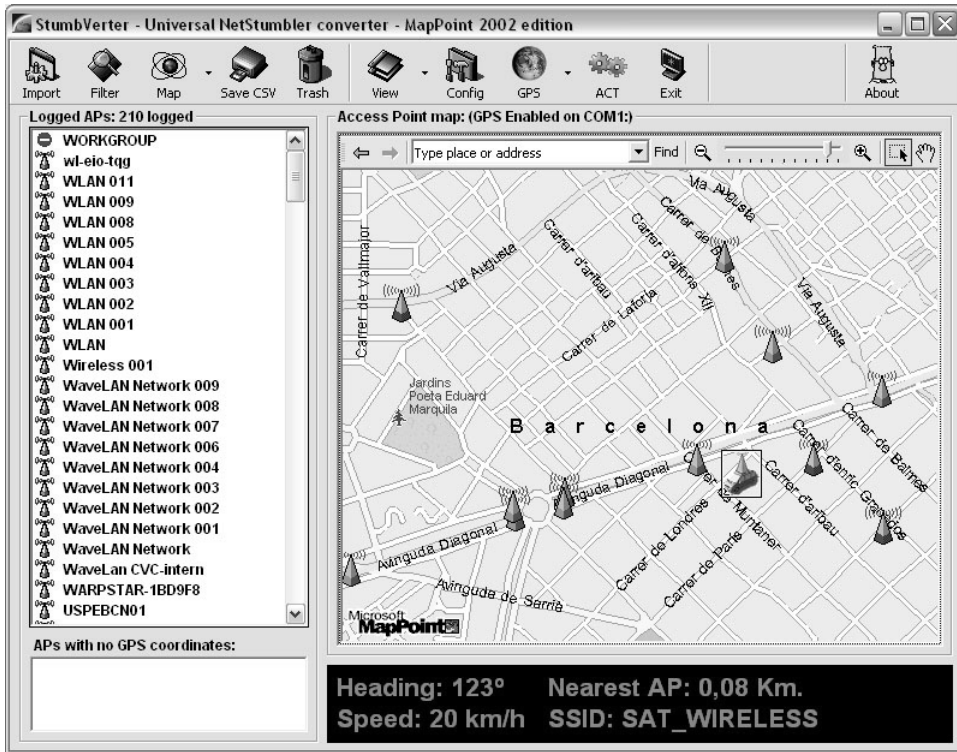


Figure 10.6 StumbVerter Map

want these for illustrative purposes. You can hide certain informational balloons if you want to show only the APs. You can also use the drawing tools to add any text, graphics, or other items to the map. When you are ready to save your map, you can either save it as a native MapPoint file or choose the CSV option if you want to save it in a text format suitable for importing into other programs.

The antenna comparison feature is useful for comparing several external antennas or different cards with built-in antennas to see which ones work best. You can import up to three different NetStumbler files, and StumbVerter grades them against the same access points and shows you the results side by side (see Figure 10.7). This can be helpful in deciding what card to use or which antennas work best if you are making one yourself.

Now that you know about some great Windows tools, I will switch platforms and talk about Linux tools. While the Windows tools are easier to install and use, there are some things that the Windows tools don't do yet, such as passive scanning and WEP cracking attempts.

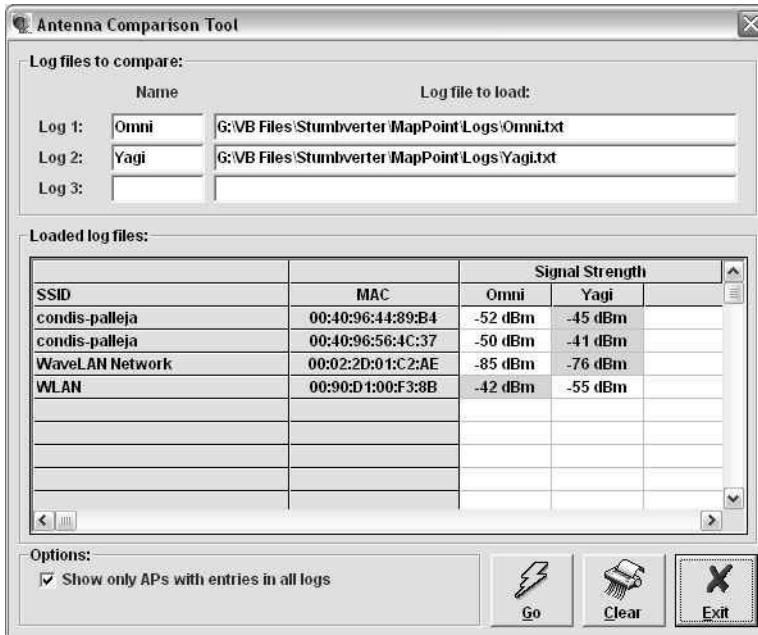


Figure 10.7 StumbVerter Antenna Comparison Screen



Kismet Wireless: A Wireless Network Discovery Program for Linux

Kismet Wireless

Author/primary contact: Mike Kershaw

Web site: www.kismetwireless.net/

Platforms: Most Linux

License: GPL

Version reviewed: .4.0.1

Mailing lists:

wireless@kismetwireless.net

Primarily for Kismet usage, suggestions, discussion, announcements of new features, and so on. Subscribe by sending an e-mail with "subscribe" in the body to wireless-subscribe@kismetwireless.net.

There is also an archive of past discussions at www.kismetwireless.net/archive.php.

wireless-security@kismetwireless.net

A mailing list for discussion of wireless security, vulnerabilities, and other topics not directly related to Kismet. Subscribe by sending an e-mail with "subscribe" in the body to wireless-security-subscribe@kismetwireless.net.

Kismet Wireless is one of the leading wireless sniffers for the Linux operating system. There are several programs, including AeroSniff and Prism2Dump, that work well on Linux as well. I chose to review Kismet because of its growing support base and add-on modules in addition to its support for a wide variety of wireless hardware. It is also a client-server tool like Nessus, which gives it even more flexibility.

Another nice thing about using the Linux platform is that you can run WEPcrack and AirSnort, which are Linux-only programs right now. As of publication, there weren't any really good WEP testing open source software available for the Windows platform, though I expect this to change.

Kismet has some features that go beyond the basic functionality of a program like NetStumbler. Kismet works with a number of other programs and can be designed to gather weak encryption keys for cracking attempts by external programs. You can even run Kismet in IDS mode to look for intrusion attempts coming from your wireless network.

Installing Your Network Interface Card and Drivers

Before loading Kismet, you should make sure your card supports it. Kismet currently works with the following wireless cards:

- D-Link
- Linksys (PCI and PCMCIA only)
- RangeLan
- Cisco Aeronet
- ORiNOCO

Theoretically, Kismet should work with any card that uses the Prism II and Hermes chipsets or ones that can be put into rf_mon or Monitor mode, but your results may vary. I recommend that you stick with one of the above cards for the fewest problems.

Now the fun really begins. There are several steps to getting your Linux system ready to be a wireless sniffer. These steps will vary slightly depending if you have a different hardware and software configuration than the procedure. Check the documentation on the Kismet Web site to see if there are specific instructions for your hardware.

1. Start by making sure your PCMCIA drivers are up to date (assuming your card uses the PCMCIA card slot). If you have installed a fairly recent version Linux, then you are probably okay. This installation example uses Mandrake Linux 9.1.
2. If you need the latest drivers, go to www.rpmfind.com and search for the file `pcmcia-cs` for your distribution. Run the RPM and it will install the latest drivers.
3. Make sure you have all the correct wireless drivers loaded for your card.
Wireless drivers for Linux are not quite as well supported as those for Windows and don't usually have a nice graphical interface to install them. (Hopefully this will change as vendors add support for Linux and someone produces RPMs for installing the drivers.)

I had to “roll my own” drivers, and the experience was less than fun. If possible, pick one of the supported cards; there are detailed instructions and lots of information online about them. With the ORiNOCO card, I compiled the driver located on the disk that came with the card. The latest driver is also available at www.orinocowireless.com, and several other sites offer cards based on this chipset.

If you are using a Prism II card, you need the Linux wlan-ng drivers. They are available at www.linux-wlan.org/.

4. Install the drivers and any patches needed for your card to operate in the Monitor mode required by wireless sniffers. This mode is similar to the Promiscuous mode on Ethernet cards that sets the card to listen to the airwaves without associating it to a particular access point.

The following instructions are for the ORiNOCO card, which required the Monitor mode patch. Consult your documentation or the Internet for other cards.

- a. Download the file or copy it from the book’s CD-ROM.
- b. To begin the installation process, type:

```
make config
```

The configuration script asks you some basic questions about your system. The defaults are generally the correct setting.

- c. Type the following commands as root:

```
./Build
./Install
```

- d. With the ORiNOCO card, you also have to install a patch on top of this in order for it to work in Monitor mode. This may not be necessary with other cards. You can get the patch from airsnort.shmoo.com/orinocoinfo.html.
- e. If you need to patch your driver, download the patch file, otherwise go to Step 5.
- f. Untar it, and type the following commands:

```
patch -p0 < patchfile.diff
```

where you replace *patchfile.diff* with the name of the current patch file. It should write over any files that are not updated. If the `-p0` switch doesn’t work, try `-p1`.

5. Next, go into the wireless configuration file and edit the setup parameters. This file is found in `/etc/pcmcia/config.opts`.
 - If you are going to be using this card with Kismet, leave these parameters blank.
 - If you want to use it to access your local access point, enter the appropriate settings for your network in this file, such as SSID and so on.
6. You can now reboot your system with your wireless card in the slot.

When it comes up you should hear two beeps. This indicates that the network card was recognized and configured.

If you don’t hear the beeps, refer back to your card’s documentation and make sure you followed all the steps correctly.



7. Type `ifconfig` at the command prompt. You should see a `wlan01` interface. If you don't see this interface, refer back to your card's documentation and make sure you followed all the steps correctly.
8. Once you have the drivers loaded, make sure your wireless card is actually working. You should be able to get Internet access or ping a network machine on the wired LAN. If you can't, then you need to refer back to your card's installation instructions. The card must be functional before loading the Kismet software.
9. You also need to have a recent `libpcap` library available so the operating system can read packets directly from your card. Many of the tools described earlier in this book use this driver, but if you haven't loaded it yet, download it from the book's CD-ROM or www.tcpdump.org and install it.

You have now finished installing your network interface card and the drivers you need to run Kismet.

Installing Kismet

If you made it through all that unscathed, you are ready to actually load the program.

1. Download Kismet from the book's CD-ROM or the Web site.
2. Unpack the distribution.
3. Enter the following command with any appropriate configure statement(s) listed in Table 10.4 to compile Kismet:

```
./configure
```

Table 10.4 Kismet Configuration Switches

Switches	Descriptions
<code>--disable-curses</code>	Disables the curses user interface.
<code>--disable-panel</code>	Disables ncurses panel extensions.
<code>--disable-gps</code>	Disables GPS support.
<code>--disable-netlink</code>	Disables Linux NetLink socket capture (prism2/orinoco patched).
<code>--disable-wireless</code>	Disables Linux kernel wireless extensions.

(continues)

Table 10.4 Kismet Configuration Switches (*continued*)

Switches	Descriptions
--disable-pcap	Disables <code>libpcap</code> capture support.
--enable-syspcap	Uses system <code>libpcap</code> (not recommended).
--disable-setuid	Disables <code>suid</code> capabilities (not recommended).
--enable-wsp100	Enables WSP100 remote sensor capture device.
--enable-zaurus	Enables some extra stuff (like piezzo buzzer) for Zaurus PDA.
--enable-local-dumper	Forces the use of local dumper code even if Ethereal is present.
--with-ethereal=DIR	Supports Ethereal wiretap for logs.
--without-ethereal	Disables support for Ethereal wiretap
--enable-acpi	Enables Linux kernel ACPI support.

These are compile-time switches you can enter with your configure statement to enable or disable certain functions.

4. Once the configuration process completes, run the following commands as root to finish the compilation process and install the program:

```
make dep
make
make install
```

5. Once Kismet is installed, find the file `kismet.conf`, which should be in `/usr/local/etc` by default. This is where you set up your logging and interface preferences. Table 10.5 describes the parameters you can set.
6. Next, edit the file `kismet_ui.conf`, also found in `/usr/local/etc`. This sets certain interface settings. Table 10.6 lists the options.
7. Save these two files.

You are ready to start using Kismet to audit your wireless network.

Table 10.5 Kismet Logging and Interface Options

Parameters	Descriptions
Capture source	Defines what interfaces Kismet will listen on. Normally your main wireless interface (wlan0) should already be set up here. If you want to add additional interfaces, do it in the format: <code>source=type,interface,name</code> . For example, <code>source=prism2,wlan0,Prism</code> directs Kismet to listen on wlan0 for a prism2 type card. This shows up as <code>Prism</code> in your logs.
Fuzzy encryption	Shows any identified packets as unencrypted for those stations using undefined or proprietary encryption methods. Generally leave this off unless your card is reporting known encrypted networks as unencrypted.
Filtering packet logs	Limits what packets get logged. Use the <code>noiselog</code> option to drop any packets that seem to be broken or fragmented due to noise. In a crowded area with lots of interference or when using a card that does not have an external antenna, this can keep your log size down. The <code>beaconlog</code> option drops all but the first beacon packet from a particular access point. The <code>phylog</code> setting drops any physical layer packets that are sometimes picked up. You can use any combination of these settings.
Decrypt WEP keys	Decrypts intercepted data packets on the fly. You must first, however, have the key, which can sometimes be obtained using AirSnort (described later in this chapter). Each access point needs a separate statement in the format <code>bssid:key</code> where <code>bssid</code> is the MAC address of the access point and <code>key</code> is the key for that access point.
Using an external IDS	Sends packets to an external intrusion detection system for further analysis. You specify a <code>FIFO</code> pipe in this statement and then direct your NIDS to read from the pipe name.

Table 10.6 Kismet Interface Settings

Settings	Descriptions
Columns	Changes what columns appear in the Kismet interface and in what order. Change the value of <code>columns</code> or <code>clientcolumns</code> to what you want to see. A complete listing of the columns available is in the Kismet man pages.
Colors	Changes the colors of any of the elements of the display. Change the <code>colorxxx</code> setting to the color code you want. You will have to play with it a bit to get the colors right. (I found the defaults to be acceptable except for printing, and changed those to a more printer-friendly color.)

Using Kismet Wireless

Start Kismet by running the executable file from the command line or from an X-Windows terminal that supports the Curses toolkit. The main interface displays (see Figure 10.8). Kismet immediately starts reporting any wireless networks in your area and information on them.

The interface is divided into three main sections. The Network List section on the left shows all the currently active wireless networks that Kismet can see and some basic information on them: the SSID of the network (if available), the type (access point versus

```

dragorn@qtr.lan.nerv-un.nett:/home/dragon
Network List—(Autofit)—
Name          T W Ch Packts Flags  Data Clnt
p@thf1nd3r    A Y 06   171          70  35
<no ssid>     A N 05     1           0   0
KrullNet1     A Y 06    27           0   0
linksys       A N 06    81 FU4        8   2
marley        A N 06   312          17   1
<no ssid>     D N --    20 A2        20  18
! PARMAS      A N 07    30           0   0
<no ssid>     A Y 06     1           0   0
GRXWirelessNetwork A Y 06     2           0   0
! SECMAS      A N 07    13           0   0
<no ssid>     D N --     1 A4         1  66
! <Lucent Outdoor Router> D N --   267          267  1

Info
Ntrwks      105
Pckets     1258
Cryptd      104
Weak         0
Noise      289
Discrd      289
Pkts/s       50

Elapspd
000027

Status
Found IP 159.139.90.1 for <no ssid>::00:04:76:BB:A7:04 via ARP
Found IP 159.139.90.1 for <no ssid>::00:04:76:BB:A7:04 via ARP
Found IP 159.139.90.1 for <no ssid>::00:04:76:BB:A7:04 via ARP
Found IP 159.139.120.13 for <no ssid>::00:B0:D0:DE:60:E3 via TCP
Battery: AC charging 100% 0h0m0s

```

Figure 10.8 Kismet Wireless Main Screen

node), whether or not it is encrypted using WEP, the channel it is broadcasting on, the number of packets intercepted so far, any flags on the data, and the amount of data going through the network. The display is color coded with active networks appearing in red and ones that are no longer active in black.

The Info box on the right shows overall statistics for this capture session, including the total number of networks sensed, the total number of packets, the number of packets that were encrypted, weak networks perceived, packets with a high noise level, packets that were discarded, and the average number of packets per second.

The Status box on the bottom contains a scrolling view of events as they happen. Messages pop up when new networks appear or other events happen.

Because Kismet is a command line tool, albeit with a GUI, it uses key commands to control its functions. Table 10.7 lists the key commands available from the main screen.

Table 10.7 Kismet Key Commands

Key Commands	Descriptions
a	Shows statistics about packet counts and channel allocation.
c	Opens a client pop-up window to display clients in the selected network.
d	Instructs the server to start extracting printable strings from the packet stream and displays them.
e	Opens a pop-up window on Kismet servers. This lets you simultaneously monitor two or more Kismet servers on different hosts (remember, it's a client-server architecture).
f	Follows the estimated center of a network and displays a compass.
g	Groups currently tagged networks.
h	Gets a listing of possible commands.
i	Displays detailed information about the current network or group.
l	Shows signal/power/noise levels if the card reports them.

(continues)

Table 10.7 Kismet Key Commands (*continued*)

Key Commands	Descriptions
m	Mutes sound and speech if they are enabled (or turns them on if they were previously silenced). You must have sound or speech enabled in your configuration to be able to use them.
n	Renames the selected network or group.
p	Displays packet types as they are received.
r	Displays a bar graph of the packet rate.
s	Sorts the network list differently.
t	Tags (or untags) the current network.
u	Ungroups the current network.
w	Displays all previous alerts and warnings.
z	Zooms the network display panel to full screen (or returns it to normal size if it is already zoomed).

As noted above, you can expand views of information on each network detected to show all the details on a particular access point by entering `i` at the command line. Figure 10.9 illustrates this output.

You can also expand the network box to full screen and see additional information on each network, such as the manufacturer of the equipment detected using the `z` command. This may make it easier to organize your access points into groups if you are trying to track a particular set of APs and want to be able to filter the others out. Do this with the `g` and `u` commands to group and ungroup, respectively.

The sound feature is handy—it beeps when you detect new networks. You can toggle that option off using the `m` command if you are going in and out of many network's reception areas. Otherwise you get a cacophony of beeps!

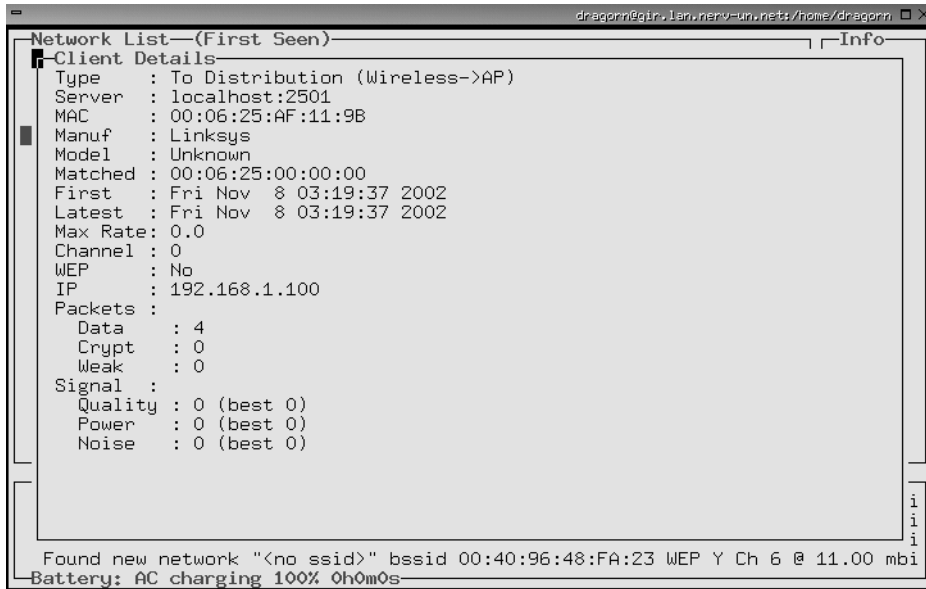


Figure 10.9 Kismet Network Detail Screen

Kismet GPS Support

Kismet has the ability to record GPS data if you have a GPS receiver plugged into your machine. You need the GPS daemon software GPSD for Kismet to read it. You can get GPSD at <http://russnelson.com/gpsd/>. You must enable GPS support when compiling Kismet using the compile-time parameters in Table 10.4. Kismet then automatically picks up the coordinates of any networks sensed and logs them.

You can take this one step further and map these coordinates just like with the Windows program. Kismet comes with a built in program called GPSMAP that automatically plots the data collected onto maps in .gps format. The downside is you have to provide your own GPS-calibrated map. There is an open source mapping program for Linux called GPSTDrive, which you can download from <http://gpsdrive.kraftvoll.at/index.shtml>.

Kismet IDS

You can also set up Kismet as a wireless IDS. Kismet will intercept incoming signals and detect wireless traffic that is known to be associated with war driving or other suspicious wireless activity. It detects about 10 different kinds of traffic, including NetStumbler polls and activity from Airjack and other wireless hacking tools. Currently this IDS capability is fairly limited, but expect it to expand in the future. And, since it's open source, you can always expand it yourself by writing your own alerts. You can also pipe your Kismet data through a traditional IDS such as Snort for more detailed analysis. The IDS feature is set

in `kismet.conf` and is turned off by default. You can also set up Kismet to gather known cryptographically weak keys for a program such as AirSnort, the next tool in this chapter, which analyzes wireless packets and attempts to crack the WEP encryption.



AirSnort: A WEP Encryption Key Recovery Program

AirSnort	
Original authors/primary contacts:	Jeremy Bruestle and Blake Hegerle
Web site:	http://schmoo.airsnort.org
Platforms:	Most Linux
License:	GPL
Version reviewed:	2.4.22

The authors developed AirSnort as a practical application to demonstrate the weakness in the WEP, the wireless encryption protocol. A paper entitled “Weaknesses in the Key Scheduling Algorithm of RC4,” written by the cryptographic experts Fluhrer, Martin, and Shamir, detailed a theoretical weakness in the WEP algorithm, describing how some of the Initialization Vectors (IVs) were weak. Packets encrypted with these weak IVs could be collected and eventually enough data would be present to extrapolate the shared secret key. This allowed the packets to be easily decrypted. Two tools were released shortly thereafter, AirSnort and WEPCrack, that employed the described weakness to recover WEP keys, effectively cracking WEP. They are both good tools, but AirSnort has some additional functionality as a wireless sniffer. AirSnort is now an open source project hosted on SourceForge.net and has been extended and improved considerably since its release. Given that there are no real alternatives under Windows for doing this, AirSnort and WEPCrack are currently the only viable alternatives for testing your WEP.

Uses for AirSnort

Why use AirSnort on your wireless network? Some might say there is no legitimate use for the program and its only purpose is as a hacker’s tool. However, I believe that the only way to know what the exposure on your wireless network is for *you* to do what the hackers do to see if your encryption is crackable and the amount of time it takes to do it. AirSnort lets you do just that.

By attempting to crack your wireless encryption, you can see if it is crackable. If you are using standard WEP, then it is merely a matter of time. It is a mathematical certainty that it can be cracked at some point using this tool. The question is, how long does it take? If it’s a very long time, you can reasonably assume you are pretty safe. If the traffic level on your wireless LAN is small, then it might take days or even weeks. This puts your network out of the realm of practicality of most casual hackers. However, if it’s a busy network, then someone might be able to pick up enough packets to break it in a few hours or a day.

Knowing this will help you to better protect your network. It can justify putting in further protections, such as better physical controls or limiting the traffic on that network. It also might justify upgrading your wireless equipment. Cisco Aeronet gear uses a variation of WEP called LEAP to improve and fix the weakness with the original WEP protocol. A wireless network using that protocol should be uncrackable, at least with readily available tools. You may find that your traffic level doesn't make it practical to crack your encryption. Either way, you'll sleep better at night knowing.

Installing AirSnort

Getting the drivers and software working for AirSnort can be quite a chore. Its requirements closely match those of the Kismet program. Refer back to the "Installing Your Network Interface Card and Drivers" section and follow that procedure. Finally, when all the moons align and you get all these things in order, you are ready to install the program. This is the easy part.



1. Download the program file from the book's CD-ROM or the official Web site and unzip it.
2. Change to the directory where you unzipped the file and run the script

```
./autogen.sh
```

3. Become root and then run

```
make
```

The program will be built for you automatically. If this completes without errors, you have successfully installed AirSnort.

Running AirSnort

AirSnort has three primary executable files.

- **airsnort** does the work of collecting the packets from some source, usually your wireless network card.
- **gensnort** sorts through the captured data for weak keys.
- **decrypt** does offline decryption attempts for files loaded from another source.

AirSnort accept files from other wireless sniffers as long as they are saved in pcap format. Kismet, our Linux wireless tool of choice, will specifically pull out interesting packets for AirSnort ahead of time, saving this step.

You don't have to do all the data collection at once. AirSnort can save a session and let you open it later and add to it. This makes AirSnort a particularly dangerous tool to wireless networks, because someone doesn't have to spend a single uninterrupted session near your facility to collect enough packets to crack your network. They can split their collection activities into smaller, less noticeable time increments, assuming the target network doesn't change its keys often.

Once you have AirSnort installed, you can start it by typing `airsnort` at the command line. The interface is simplicity itself: it is a single screen that shows the interesting packets and the total number of encrypted and unencrypted packets. The top section shows you settings such as NIC card type and so forth. On the left, you can change some settings, such as the **breadth**—the number of guessing attempts AirSnort will make for each key byte—for either 40-bit or 128-bit decryption attempts. The default is 3 for 40-bit encryption and 2 for 128-bit encryption. If you don't have a lot of data or you have a lot of excess processing power, you can try increasing this slightly, but don't go much more than 4 or 5.

After that, it is time to just sit back and collect packets. Don't expect to be able to crack WEP keys in just a few moments. For AirSnort to work properly, it needs approximately 1,500 to 4,500 packets with weak keys. This amounts to between 100MB and 500MB of data. On a moderately busy network, it might take a day or more to collect this much data. On slower networks it could take much longer and on busier networks much less. Expect it to take at least a couple of hours but probably longer. Of course, all of this is based on a little luck too, so your results may vary from an hour to never. Generally, you want to spend about as much time collecting data as you think the average outsider might be able to spend undetected. And of course, AirSnort's resume session feature could make this time window much shorter since they could use multiple collection sessions.

When a successful crack of the WEP key has occurred, it appears in both plain text and the original hexadecimal on the far left of the display and the capture session ends. Happy WEP cracking!

What do you do if you find your WEP keys? Well, don't panic, because most casual hackers won't go to the trouble. However, you should think about taking steps to increase the security of your wireless network to make it harder for outsiders to collect this data. There are a number of steps you can take, ranging from replacing your equipment to reconfiguring and changing your AP position. You will have to decide based on the sensitivity of the data on your network which ones are appropriate.

Steps for More Secure Wireless LANs

The chances are that eventually you will have to implement wireless technology. Even if you don't, you should still occasionally audit your network and make sure someone isn't running a rogue wireless access point. While running any wireless access is a risk, you can lessen your exposure by taking the following preventative measures.

Turn On WEP

By encrypting your data you are requiring hackers to spend a lot more time and effort to get to your wireless data and network. This will discourage casual hackers and make the serious ones have to hang around your area for a day or so, increasing the chances that they will be noticed by security personnel or vigilant employees.

Use Wireless Equipment with an Improved Encryption Protocol

As mentioned earlier, Cisco equipment uses an improved version of WEP call LEAP, which so far has proven impervious to cracking attempts. There is also a new standard, 802.11i, which permanently fixes the problems with WEP. Unfortunately, 802.11i has only recently been approved as a standard and equipment based on it should be available soon. If you can get them, do so. The pricing shouldn't be any different than the older 802.11a and 802.11b gear.

Require Wireless Users to Come in Via a VPN Tunnel

This step adds a mostly insurmountable hurdle for would-be wireless intruders. Even if they manage to crack your WEP encryption, they then have to tackle the VPN encryption. Some vendors (such as SonicWALL with its Wi-FiSec feature) have added this capability into their equipment. The downsides are that there is an additional layer of complexity for your users and this makes it harder to support "guest" users, as they would need VPN client software loaded as well as the WEP key to access the WLAN.

Treat Your Wireless Network as Untrusted

Since you cannot control what traffic is coming across the air to access points, you shouldn't treat it any differently than the public side of your firewall. If you can afford it, place a firewall between your wireless network and your LAN (see Chapter 3 for some open source options) or place it on your DMZ. Then you can filter certain kinds of attack packets, limit types of traffic, and track any activity coming from that interface.

Audit Your Wireless Perimeter on a Regular Basis

This is especially important if you are in one of those dense areas mentioned earlier. Test to see how far away your signal can be picked up and if your network is overlapping nearby ones. Even if you don't officially allow wireless access, you should do this periodically to locate any rogue or "unofficial" access points. Wireless has become so cheap and easy to set up that unthinking or uncaring managers will often go to the local electronics store and set up an access point for some temporary purpose, such as a demo in an unwired conference room, opening up your network to wireless attack. Additionally, remember that a lot of new PCs, especially laptops, are coming with Wi-Fi cards built-in, and enabling them is easy to do. You may be running wireless on your network without realizing it. A wireless audit is the only way to find out.

Move Your Access Points

Sometimes just by moving the base station into an interior room you can decrease the broadcast of your wireless network signal considerably. Use your wireless audit results to figure out which access points are problematic. Play around with placement so you get

optimal reception inside the building but minimal reception outside the building. For example, if your building has a large parking lot in front and a wooded lot in back, moving the base station to the back of the building will probably still allow most internal people to reach it, but will limit the radiation of the signal to an area that is not easily accessible by war drivers.

Configure Your Wireless Network Properly

There are many features and settings you can use to increase your security considerably. Not all equipment supports these options, but here are some things you can do.

- Turn off the SSID broadcast. Doing this requires a user to know the SSID to establish a session with the base station. This acts as a weak password. However, if an eavesdropper manages to crack your encryption, he or she will be able to gain the SSID easily.
- Restrict access by MAC address. This makes it more difficult for someone to gain access to your network via a wireless base station. In most access points, you can restrict access to certain hardware MAC addresses. This is a fairly strong method of authentication, since only people with the correct serialized network card can gain access. However, it may be cumbersome for administrators to keep track of authorized NIC cards and it doesn't allow for instant access for a new user in your office. Also, if the attacker knows one of the authorized MAC addresses, it is possible to forge this address on his or her card and masquerade as that user.

Train Your Staff

As with all computer security, the human element can be your weakest or strongest point. Make sure security guards, receptionists, and other personnel know how to look for suspicious behavior associated with war driving. For example, if they see someone sitting in your parking lot for long periods of time, possibly with a strange antenna on their roof, then it might be likely he or she is targeting your wireless network.

Also, develop and get approval on a company-wide policy for deploying wireless LANs. Make sure managers know that they can't set up a wireless LAN themselves; that they need to go through you for an official connection. Make them understand how they are putting the whole company at risk with this behavior. Sometimes a demonstration is the best way to get the danger of this across. An informed workforce can be your best defense.