the technical editor for *Hacking For Dummies* and *Norton Internet Security For Dummies*. Peter is listed in the *International Who's Who of Professionals*. In addition, he was only the third editor in the three-decade history of *EDPACS*, a publication in the field of security, audit, and control. He finds time to be a part-time lecturer in data communications at Seneca College (`http://cs.senecac.on.ca`). He lives with his wife Janet, daughter Kelly, two cats, and a dog in Toronto, Ontario.

Dedication

Little G — this one's for you. You're such a great motivator and inspiration to me — more than words can say. Thanks for reminding me of what's really important. Thanks for being you.

—Kevin

To all my friends and enemies. Hopefully the first group is bigger than the second.

—Peter

Authors' Acknowledgments

Kevin:

Thanks to Melody Layne, our acquisitions editor, for approaching me about this project and getting the ball rolling.

I'd like to thank our project editor, Chris Morris, as well as Kevin Kirschner and all the behind-the-scenes copy editors for pulling this thing together. Many thanks to my co-author Peter T. Davis for working with me on this book. It has been an honor and a pleasure.

I'd also like to thank Hugh Pepper, our technical editor, for the feedback and insight he gave us during the technical editing process.

Also, many thanks to Devin Akin with Planet3 Wireless for writing the foreword. Major kudos too for all the positive things you've done for the industry with the CWNP program. You're a true wireless network pioneer.

Many thanks to Ronnie Holland with WildPackets, Chia Chee Kuan with AirMagnet, Michael Berg with TamoSoft, Matt Foster with BLADE Software, Ashish Mistry with AirDefense, and Wayne Burkan with Interlink Networks for helping out with my requests.

One of the most difficult tasks for a consultant today is teaching customers about wireless LAN technology. Often, organizations understand neither the technology nor the risks associated with it. 802.11 networks have a significant ROI for some organizations, but inherently create a security hole so big that you could drive a truck through it. Organizations should carefully consider whether 802.11 networks are feasible and can be cost-justified. Many things go into the securing of 802.11 networks, from secure installation to end-user and IT staff training.

Forgetting to cover a single base in wireless-LAN security can lead to intrusion and financial disaster. The risks can often far outweigh the gain of using 802.11 technology, so organizations decide to have a no-use policy. Still, those organizations *must* consider how to protect from wireless intrusion. One of the tricks to getting customers to "bite" — commit to the notion of protecting their wireless LAN — is to give them a quick demonstration of hacking tools. If they have (for example) a heavily loaded 802.11g network secured with WEP, cracking their WEP key should open their eyes very quickly.

Keep in mind that these demonstrations should ALWAYS be done with the permission of a person in authority at the client organization — and in a closed environment. Doing otherwise can lead to criminal prosecution, defamation of your organization, and a plethora of other undesirable results.

Time is never the IT professional's friend. Staying abreast of the latest tools and techniques takes lots of hard work and time. Reading a book like this one is a worthy endeavor toward becoming an experienced wireless security professional.

I am a firm believer in picking a field of study and becoming the best you can be in that particular area. Wireless LAN technology is so deep and wide that it can easily consume all of your time, so focusing on being a wireless LAN *security* professional is a reasonable and attainable choice. The market demand, the pay, and the career itself are all good. Best wishes to all who choose this career path — or endeavor to increase their networking knowledge by reading great books like this one.

Devin K. Akin

Chief Technology Officer, The Certified Wireless Network Professional (CWNP) Program `http://www.cwnp.com`

Introduction

Welcome to *Hacking Wireless Networks For Dummies.* This book outlines plain-English, wireless-network hacker tricks and techniques you can use to ethically hack 802.11-based wireless networks (yours or someone else's if you've been given permission) and discover security vulnerabilities. By turning the tables and using ethical hacking techniques, you then have a leg up on the malicious hackers — you'll be aware of any vulnerabilities that exist and be able to plug the holes before the bad guy has a chance to exploit them.

When we refer to *ethical hacking,* we mean the professional, aboveboard, and legal type of security testing that you — as an IT professional — can perform as part of your job. Villains need not apply.

Wireless networks are popping up everywhere. They provide a lot of freedom but not without cost. All too many wireless networks are left wide open for attack. As with any other computer or network, you must be up on the latest security concepts to properly secure 802.11-based wireless networks. But locking them down involves more than just port-scanning testing and patching vulnerabilities. You must also have the right security tools, use the proper testing techniques, and possess a watchful eye. And *know your enemy:* It's critical to think like a hacker to get a true sense of how secure your information really is.

Ethical hacking is a means of using the bad-guy (black-hat) techniques for good-guy (white-hat) purposes. It's testing your information systems with the goal of making them more secure — and keeping them that way. This type of security testing is sometimes called *penetration testing, white-hat hacking,* or *vulnerability testing,* but it goes further than that as you'll see when we outline the methodology in this book.

If you use the resources provided in this book, maintain a security-focused mindset, and dedicate some time for testing, we believe you'll be well on your way to finding the weaknesses in your wireless systems and implementing countermeasures to keep the bad guys off your airwaves and out of your business.

The ethical hacking tests and system-hardening tips outlined in this book can help you test and protect your wireless networks at places like warehouses, coffee shops, your office building, your customer sites, and even at your house.

Where to Go from Here

The more you know about how the bad guys work, how your wireless networks are exposed to the world, and how to test your wireless systems for vulnerabilities, the more secure your information will be. This book provides a solid foundation for developing and maintaining a professional ethical-hacking program to keep your wireless systems in check.

Remember that there's no one best way to test your systems because everyone's network is different. If you practice regularly, you'll find a routine that works best for you. Don't forget to keep up with the latest hacker tricks and wireless-network vulnerabilities. That's the best way to hone your skills and stay on top of your game. Be ethical, be methodical, and be safe — happy hacking!

With the convenience, cost savings, and productivity gains of wireless networks come a whole slew of security risks. These aren't the common security issues, such as spyware, weak passwords, and missing patches. Those weaknesses still exist; however, networking without wires introduces a whole new set of vulnerabilities from an entirely different perspective.

This brings us to the concept of ethical hacking. *Ethical hacking* — sometimes referred to as *white-hat hacking* — means the use of hacking to test and improve defenses against *un*ethical hackers. It's often compared to penetration testing and vulnerability testing, but it goes even deeper. Ethical hacking involves using the same tools and techniques the bad guys use, but it also involves extensive up-front planning, a group of specific tools, complex testing methodologies, and sufficient follow-up to fix any problems before the bad guys — the black- and gray-hat hackers — find and exploit them.

Understanding the various threats and vulnerabilities associated with 802.11-based wireless networks — and ethically hacking them to make them more secure — is what this book is all about. Please join in on the fun.

In this chapter, we'll take a look at common threats and vulnerabilities associated with wireless networks. We'll also introduce you to some essential wireless security tools and tests you should run in order to strengthen your airwaves.

Why You Need to Test Your Wireless Systems

Wireless networks have been notoriously insecure since the early days of the 802.11b standard of the late 1990s. Since the standard's inception, major 802.11 weaknesses, such as physical security weaknesses, encryption flaws, and authentication problems, have been discovered. Wireless attacks have been on the rise ever since. The problem has gotten so bad that two wireless security standards have emerged to help fight back at the attackers:

- **Wi-Fi Protected Access (WPA):** This standard, which was developed by the Wi-Fi Alliance, served as an interim fix to the well-known WEP vulnerabilities until the IEEE came out with the 802.11i standard.
- **IEEE 802.11i (referred to as WPA2):** This is the official IEEE standard, which incorporates the WPA fixes for WEP along with other encryption and authentication mechanisms to further secure wireless networks.

These standards have resolved many known security vulnerabilities of the 802.11a/b/g protocols. As with most security standards, the problem with these wireless security solutions is not that the solutions don't work — it's that many network administrators are resistant to change and don't fully implement them. Many administrators don't want to reconfigure their existing wireless systems

and don't want to have to implement new security mechanisms for fear of making their networks more difficult to manage. These are legitimate concerns, but they leave many wireless networks vulnerable and waiting to be compromised.

Even after you have implemented WPA, WPA2, and the various other wireless protection techniques described in this book, your network may still be at risk. This can happen when (for example) employees install unsecured wireless access points or gateways on your network without you knowing about it. In our experience — even with all the wireless security standards and vendor solutions available — the majority of systems are still wide open to attack. Bottom line: Ethical hacking isn't a do-it-once-and-forget-it measure. It's like an antivirus upgrade — you have to do it again from time to time.

Knowing the dangers your systems face

Before we get too deep into the ethical-hacking process, it will help to define a couple of terms that we'll be using throughout this book. They are as follows:

- **Threat:** A *threat* is an indication of intent to cause disruption within an information system. Some examples of threat agents are hackers, disgruntled employees, and malicious software (malware) such as viruses or spyware that can wreak havoc on a wireless network.
- **Vulnerability:** A *vulnerability* is a weakness within an information system that can be exploited by a threat. Some examples are wireless networks not using encryption, weak passwords on wireless access points or APs (which is the central hub for a set of wireless computers), and an AP sending wireless signals outside the building. Wireless-network vulnerabilities are what we'll be seeking out in this book.

Beyond these basics, quite a few things can happen when a threat actually exploits the vulnerabilities of a various wireless network. This situation is called *risk*. Even when you think there's nothing going across your wireless network that a hacker would want — or you figure the likelihood of something bad happening is very low — there's still ample opportunity for trouble. Risks associated with vulnerable wireless networks include

- Full access to files being transmitted or even sitting on the server
- Stolen passwords
- Intercepted e-mails
- Back-door entry points into your wired network
- Denial-of-service attacks causing downtime and productivity losses
- Violations of state, federal, or international laws and regulations relating to privacy, corporate financial reporting, and more

Network attacks

When it comes to the nitty-gritty bits and bytes, there are a lot of techniques the bad guys can use to break inside your wireless realm or at least leave it limping along in a nonworking state. Network-based attacks include

- Installing rogue wireless APs and "tricking" wireless clients into connecting to them
- Capturing data off the network from a distance by walking around, driving by, or flying overhead
- Attacking the networking transactions by spoofing MAC addresses (masquerading as a legitimate wireless user), setting up man-in-the-middle (inserting a wireless system between an AP and wireless client) attacks, and more
- Exploiting network protocols such as SNMP
- Performing denial-of-service (DoS) attacks
- Jamming RF signals

Software attacks

As if the security problems with the 802.11 protocol weren't enough, we now have to worry about the operating systems and applications on wireless-client machines being vulnerable to attack. Here are some examples of software attacks:

- Hacking the operating system and other applications on wireless-client machines
- Breaking in via default settings such as passwords and SSIDs that are easily determined
- Cracking WEP keys and tapping into the network's encryption system
- Gaining access by exploiting weak network-authentication systems

Chapter 2

The Wireless Hacking Process

In This Chapter

- Understanding the hacking process
- The Ten Commandments of Ethical Hacking
- Understanding the standards
- Evaluating your results

We teach courses on ethical hacking — and when you're teaching, you need an outline. Our teaching outline always starts with the introduction to the ethical-hacking process that comprises most of this chapter. Inevitably, when the subject of an *ethical* hacking process comes up, the class participants visibly slump into their chairs, palpable disappointment written all over their faces. They cross their arms across their chests and shuffle their feet. Some even jump up and run from class to catch up on their phone calls. Why? Well, every class wants to jump right in and learn parlor tricks they can use to amaze their friends and boss. But that takes procedure and practice. Without a defined process, you may waste time doing nonessential steps while omitting crucial ones. So bear with us for a while; this background information may seem tedious, but it's important.

Obeying the Ten Commandments of Ethical Hacking

In his book *Hacking For Dummies* (Wiley), Kevin discussed the hacker genre and ethos. In Chapter 1, he enumerated the Ethical Hacking Commandments. In that book, Kevin listed three commandments. But (as with everything in networking) the list has grown to fill the available space. Now these commandments were not brought down from Mount Sinai, but thou shalt follow these commandments shouldst thou decide to become a believer in the doctrine of ethical hacking. The Ten Commandments are

1. Thou shalt set thy goals.
2. Thou shalt plan thy work, lest thou go off course.

wear hacker gear and drink Red Bull. What you do have to do is keep plugging away until you reach your goal.

In the previous commandment we talked about acting professionally. One hallmark of professionalism is keeping adequate records to support your findings. When keeping paper or electronic notes, do the following:

- Log all work performed.
- Record all information directly into your log.
- Keep a duplicate of your log.
- Document — and date — every test.
- Keep factual records and record all work, even when you think you were not successful.

This record of your test design, outcome, and analysis is an important aspect of your work. Your records will allow you to compile the information needed for a written formal report. You should take care in compiling your records. Be diligent in your work and your documentation.

Thou shalt respect the privacy of others

Treat the information you gather with the utmost respect. You must protect the secrecy of confidential or personal information. All information you obtain during your testing — for example, encryption keys or clear text passwords — must be kept private. Don't abuse your authority; use it responsibly. This means you won't (for example) snoop into confidential corporate records or private lives. Treat the information with the same care you would give to your own personal information.

Thou shalt do no harm

The prime directive for ethical hacking is, "Do no harm." Remember that the actions you take may have unplanned repercussions. It's easy to get caught up in the gratifying work of ethical hacking. You try something, and it works, so you keep going. Unfortunately, by doing this you may easily cause an outage of some sort, or trample on someone else's rights. Resist the urge to go too far — and stick to your original plan.

Also, you must understand the nature of your tools. Far too often, people jump in and start using the tools shown in this book without truly understanding the full implications of the tool. They do not understand that setting up a monkey-in-the-middle attack, for example, creates a denial of service. Relax, take a deep breath, set your goals, plan your work, select your tools, and (oh yeah) read the documentation.

Each step has associated tasks that provide more detail and specific tests. As well, each step has a table that outlines the expected results. For example, expected results for Step 3 include these:

- Verification of the organization's security policy and practices — and those of its users.
- Identification of the outermost physical edge of the wireless network.
- Identification of the logical boundaries of the wireless network.
- Enumeration of access points that lead into the network.
- Identification of the IP-range (and possibly DHCP-server) of the wireless network.
- Identification of the encryption methods used for data transfer.
- Identification of the authentication methods of exploitable "mobile units" (that is, the clients) and users.
- Verification of the configuration of all devices.
- Determination of the flaws in hardware or software that facilitate attacks.

Obviously, you need to cut and paste these tests according to your needs. For instance, should your organization not have infrared, then you would skip Step 11.

The OSSTMM is available from `www.isecom.org/osstmm/`.

With resources like these, you have a methodology — and everything you need to map out your plan. But rather than leave you hanging there, the rest of the book shows you how to work through a methodology. In Chapter 3, you develop a methodology for a review. In Chapter 4, you select your weapons of mass disruption. Chapters 6 through 16 show you how to use the tools to test your security posture. The only thing left after that is to evaluate your results. So . . .

Chapter 4

Amassing Your War Chest

In This Chapter

- Choosing your platform: PDAs versus laptops
- Choosing your software
- Using software emulators
- Choosing transceivers, antennae, and GPS
- Signal jamming

A cyberwar is being waged. Your perimeter is under siege. What makes the attack especially insidious is that you cannot see your enemy. This isn't hand-to-hand combat. Your enemy could be 2 miles from your office and still access your network and data. Your access point is your first line of defense in this war. It behooves you, then, to prepare for battle.

One way to prepare for any war is to participate in war games. Real war games allow you to test your equipment, tactics, and operations. In this case, war games allow you to test your wireless networks under normal conditions. Like the Reservist going off to war, you also must receive adequate training on the latest weapons and tactics. Although the rest of the book focuses on tactics, this chapter focuses on equipment. You need practice with the tools the crackers use for real.

You need some hardware and software, but you have choices about what type of hardware and software you use. This chapter serves as your armory. If you favor the Windows platform, we have some tools for you. Should you favor Linux, you will find some tools as well. We don't leave Apple enthusiasts out; we have something for you, too.

11. **From the Commands panel, click Edit Virtual Machine Settings.**

    VMware presents the window shown in Figure 4-10.

12. **Click CD-ROM.**

    If you want to install the operating system from a CD, then skip to Step 14.

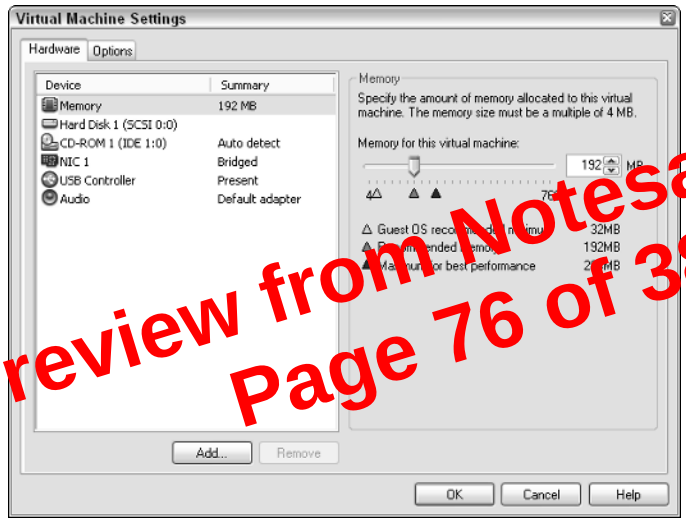

Figure 4-10: Virtual Machine settings.

13. **From the right-hand pane, select Use ISO image.**

14. **Click the Browse button and find your ISO image. Click OK.**

15. **Click Start This Virtual Machine from the left-hand pane.**

    When you do this, you see a familiar display: The VM goes through the POST routine, does a memory check, and then boots itself.

Cygwin and VMware are wonderful tools, but you need to install them on your system; they won't run any other way. If you don't want to install software on your system, you can use products like Knoppix and WarLinux that boot from a diskette or a CD.

Linux distributions on CD

The following solutions are different from the partitioning and emulation solutions discussed above. What makes them different is that you don't need to install them on your system: They boot and run completely from a CD.

Knoppix, for instance, runs from a CD based on the Linux 2.6.*x* kernel. It is a free and Open Source GNU/Linux distribution. You don't need to install

Directional vs. omnidirectional antennas

We have actually had a great deal of success using directional antennas — as opposed to using omni antennas — for wardriving. If the directional antenna (or cantenna in this case) is aimed forward toward the front of the car, signals in front of you are often acquired much earlier than they are when using the omni antenna. The cantenna can then be moved left or right, peaking the signal and pointing out the exact origin or location of the wireless access point or errant signal being tracked.

With an omni, the signal strength gets stronger only as you get closer, but you can never be sure from which direction the signal is coming without actually traveling in several directions to track the signal strength. A directional antenna provides direction as well as signal strength when trying to locate a specific target. An omni can show a larger number of signals at one time than a directional antenna, but with lower signal strength than the directional antenna provides.

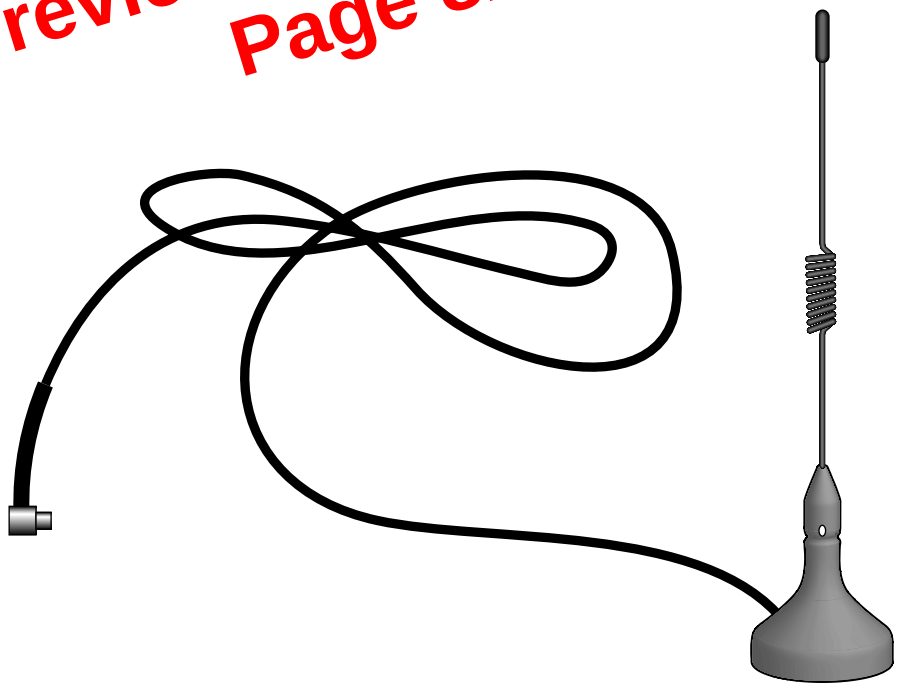

Figure 4-15: Omni-directional antenna.

The key is to look at this from a hacker's perspective. Outside of the technical methods we describe elsewhere in this book, ask yourself how a malicious outsider could gain access to your wireless network. The options and techniques are limitless.

Passive tests

The easiest way to start gathering information you can use during your social engineering tests is to simply search the Internet. You can use your favorite search engine to look up public information such as phone lists, organizational charts, network diagrams, and more. You can then see, from an outsider's perspective, what public information is available that can be used as an inroad for social engineering and ultimate penetration into your network.

One of the best tools for performing this initial reconnaissance is Google. It's amazing what you can find and find with Google. It's even more amazing that this information is made accessible to the public in the first place! You can perform generic Google queries or keywords and files that could lead to more information about your organization and network. Be sure to do both a Web and Groups search in Google because they may both contain some interesting information.

You can also perform some more advanced Google queries that are specific to your network and hosts. Simply enter the following directly into Google's search field to look for information that could be used against you:

- `site: your~public~host~name/IP` *keywords to search for*

  Look for keywords such as *wireless, address, SSID, password, .xls* (Excel spreadsheets), *.doc* (Word documents), *.ppt* (Power Point slides), *.ns1* (Network Stumbler files), *.vsd* (Visio drawings), *.pkt* (sniffer packet captures), and so on.

- `site: your~public~host~name/IP filetype:ns1 ns1`

  This searches for Network Stumbler files that contain wireless network configuration information. You can perform this query on any type of file, such as .vsd, .doc, and so on.

- `site: your~public~host~name/IP inurl:"h_wireless_11g.html"` or `inurl:"ShowEvents.shm"`

  This searches publicly accessible APs (yikes!) such as D-Link and Cisco Aironet for wireless setup pages and event logs, respectively. You may not think your systems have such a vulnerability, but do this test — you may be surprised.

- Broadcasting of SSIDs
- Admin passwords
- Remote management enabled
- Full power settings
- Use of omnidirectional antennas that come standard on most APs
- No MAC-address filtering
- WEP turned off

There are also related updates to AP firmware as well as client management software and drivers that come with the wireless systems. Wireless vendors are continually updating their firmware and software to fix security vulnerabilities and add enhanced security features, yet patching and updating is often overlooked.

Hackers know they can download the documentation for practically any 802.11-based wireless network right off the Internet. This documentation often reveals many of the default settings in use. In addition, several independent Internet sites list default settings, including:

- `www.cirt.net/cgi-bin/passwd.pl`
- `www.phenoelit.de/dpl/dpl.html`
- `http://new.remote-exploit.org/index.php/Wlan_defaults`
- `www.thetechfirm.com/wireless/ssids.htm`

If you want to see if your users or any of the systems you've set up are using vulnerable default settings, you can perform some basic tests with the information you've gathered, including

- Connecting to APs by using their default SSIDs
- Remotely connecting to the default admin port
- Spoofing MAC addresses (we cover this in detail in Chapter 13)

Refer to Chapter 8 for details of the various default setting tests you can perform against your network.

Weak Passwords

The use of weak passwords on wireless systems is a major problem. Passwords are often one of the weakest links in the information-security chain — especially on wireless networks, where they're easier to glean and crack. From remote

If you have the budget, you might want to consider using a spectrum analyzer like the ones offered by Anritsu (`www.anritsu.co.jp/E/Products/Appli/Wlan`) or Rohde & Schwarz (`www.rohde-schwarz.com`). However, some freeware spectrum analyzers are available — for example, the Waterfall Spectrum Analyzer (`http://freshmeat.net/projects/waterfallspectrumanalyzer`). A RF Spectrum Analyzer is a device that receives a chosen range of signals, in our case 2.4 GHz and 5 GHz, and displays the relative signal strength on a logarithmic display, usually a cathode ray oscilloscope.

Network Physical Security Countermeasures

Radio waves travel. This means that crackers don't need to physically attach to your network. Most likely you have locks on your doors. You might even have an alarm system to protect your physical perimeter. Unfortunately, the radio waves don't respect your perimeter security measures. Consequently, you need to walk your perimeter — whether you're an individual wanting to protect your access point or a large organization wanting to protect its wired network. While walking the perimeter, monitor the quality of the signal using the tools discussed in this chapter. When you find the signal in places where you don't want it, then turn down the power or move the access point to shape the cell shape.

Other than checking for leakage, you can monitor access points for unauthorized clients.

Checking for unauthorized users

Most access points allow you to view either the DHCP clients or the cache of MAC addresses. This is a good feature for a small network. You can review the cache from time to time to make sure that only your clients are using the access point. If you have only five clients, but you see six MAC addresses, then it just doesn't add up. After you figure out the one that doesn't belong, you can use MAC filtering to block that client.

For a large network, this feature is not very useful. Keeping track of all the MAC addresses in your organization is too difficult. As well, someone running a packet analyzer or sniffer could grab packets and get legitimate MAC addresses. A hacker could then use a MAC address changer like SMAC (`www.klcconsulting.net/smac`), which allows him to set the hardware or MAC address for any interface, say your wireless adapter or Ethernet network interface card (NIC). Figure 6-1 shows the SMAC interface. All you do is put in the hardware address you want and restart the system (or simply disable and re-enable your NIC). Your interface will have the new hardware address.

These are discussed in greater detail in the following sections. Figures 6-2 through 6-5 are simplistic depictions of the radiation patterns for the four types of antennae. Each antenna has a unique radiation pattern determined by its construction. We are limited by the print medium, so remember that the radiation pattern is three-dimensional. You may have trouble picturing this; picture a directional antenna as a conical pattern of coverage that radiates in the direction that you point the antenna, while an omnidirectional antenna's pattern of coverage is shaped more like a doughnut around the antenna.

Parabolic grid

Parabolic grid antennae are primarily used for site-to-site applications. A parabolic grid antenna may look like a satellite TV dish or like a wire grid without a solid central core. The parabolic antenna is a unidirectional antenna, meaning that it transmits in one specific direction — the direction that you point the antenna. Figure 6-2 depicts the radiation pattern of a parabolic grid antenna.

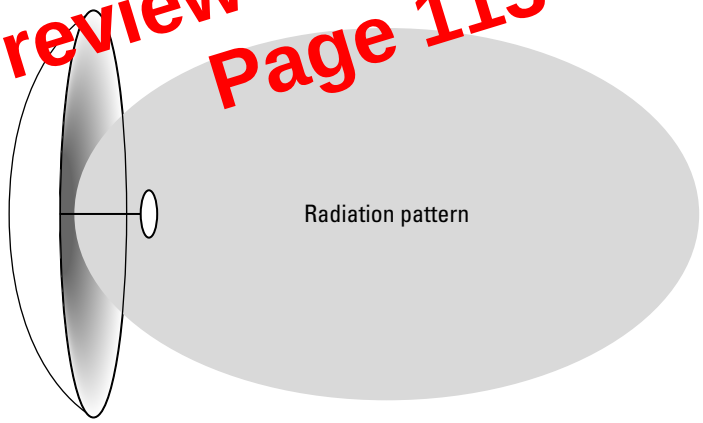

Figure 6-2: Parabolic dish or grid antenna radiation pattern.

Yagi

A yagi antenna focuses the beam, but not as much as the parabolic antenna. It's suitable for site-to-site applications in which the distance does not require a parabolic grid. Like the parabolic antenna, a yagi antenna is unidirectional. Figure 6-3 depicts the radiation pattern of a yagi antenna.

Omnidirectional

An omnidirectional antenna is one that radiates in all directions, losing power as the distance increases. Figure 6-5 depicts the radiation pattern extending in all directions outward. Many wireless base stations come with a small omnidirectional antenna.

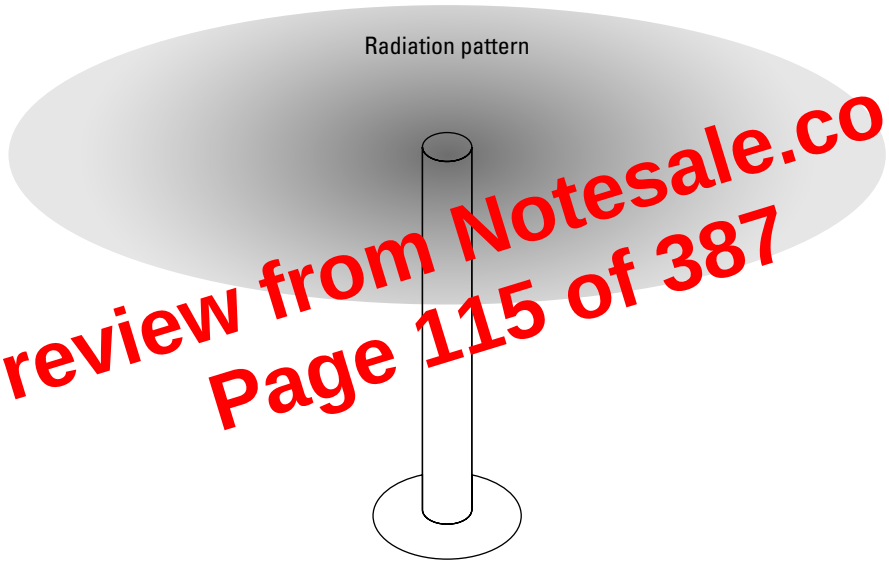

Figure 6-5 Omnidirectional antenna radiation pattern.

Adjusting your signal strength

If you find your signals are bleeding over beyond your perimeter, the first thing you need to do is to reduce the signal strength by adjusting the power settings on your access point. By doing this, you can do some cell sizing and cell shaping. Any access point not meant for the mass home market should allow you to tweak the power. Consider reducing the power of your access point to weaken the signal so that it travels a shorter distance and *doesn't* go where you *don't* want it. If you have a Cisco Aironet 340, for example, you can drop the power output from 30 mW to 5 mW.

If you adjust the power and the signal is still too strong, you need to introduce some loss through the use of an attenuator. You can pick up an attenuator at any good electronics store or find them on the Web. Coaxicom (`www.coaxicom.com`) is a good place to look for attenuators.

Chapter 7

Hacking Wireless Clients

In This Chapter

- Exploring what can happen when wireless clients are attacked
- Port scanning
- Understanding common vulnerabilities
- Undergoing basic Linux and Windows vulnerability tests
- Obtaining insecure WEP keys
- Implementing host-based defenses to help keep your network secure

This book focuses mostly on attacks against wireless *networks* as a whole — that is, 802.11-based attacks against encryption, authentication, and other protocol weaknesses. However, it's important not to forget the reason we have and use networks in the first place — our client systems. When we say *client systems,* we mean workstations, servers, and even APs that are reachable across the wireless network. If wireless networks are accessible to unauthorized people outside your organization, a lot of information can be gleaned from wireless clients. Many hacks don't even require the attacker to be authenticated to the client systems.

When you start poking around on your network, you may be surprised at how many of your wireless clients have security vulnerabilities and just what information they can reveal to attackers. That's why performing security scans on your wireless clients can be so important: It can show you what the bad guys can see if they ever are able to break through your airwaves and gain access to your network hosts.

Think like a hacker — build a mental picture of what's available to be hacked and determine methods to go about exploiting the vulnerabilities.

This chapter shows you how to test for some common wireless-client vulnerabilities. We start with how to scope out wireless hosts on the network and then move on to vulnerabilities that are specific to wireless hosts. We also outline some practical countermeasures, so you can make sure that your systems are secure.

For an in-depth look at detailed vulnerabilities across various wireless client operating systems, e-mail, malware, and more, be sure to check out Kevin's book *Hacking For Dummies* (Wiley).

What Can Happen

If your wireless systems are breached and a hacker is able to obtain access to your internal computers, several bad things can happen. First off, the attacker can gather information about your systems and their configuration, which can lead to further attacks. Such information includes:

- Open ports and available services
- Weak passwords
- WEP keys that are stored locally and not properly secured
- Acceptable usage policies and banner page information
- Operating system, application, and firmware versions returned via banners, error messages, or unique system fingerprints
- Operating system and application configuration information

The exposure of this information can lead to bigger problems such as:

- Leakage of confidential information, including files being copied and private information such as social security numbers and credit-card numbers being stolen
- Passwords being cracked and used to carry out other attacks
- Servers being shut down, rebooted, or taken completely offline
- Entire databases being copied, corrupted, or deleted

If you discover a surprising number of vulnerabilities in your wireless APs, workstations, and servers (and you likely will), don't panic. Start by addressing the issues with your most critical systems that will give you the highest payoff once secured.

- MAC addresses of the hosts found
- Services or applications that the hosts may be running
- Unauthorized hosts or applications

The big-picture view from port scanners often uncovers security issues that may otherwise go unnoticed. Port scanners are easy to use and can test systems regardless of what operating systems and applications are running. The tests can be performed very quickly without having to touch individual network hosts, which would be a real pain otherwise.

A good way to get a quick overview of which systems are alive and kicking on the network is to perform a *ping sweep.* A ping sweep is when you send out ping requests (that is, ICMP echo requests) and see if echo replies are received back. Free port scanner programs such as Foundstone's SuperScan (`www.foundstone.com/resources/proddesc/superscan.htm`) and SoftPerfect's Network Scanner (`www.softperfect.com/products/networkscanner`), as shown in Figure 7-1, often have ping sweep capabilities built in, and are all you need to get started.

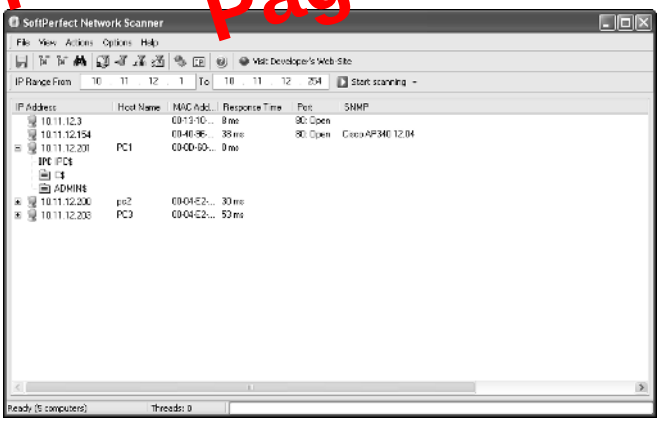

Figure 7-1: Using SoftPerfect's Network Scanner to find live wireless hosts.

Network Scanner also performs ARP lookups and displays each host's MAC address. This capability is especially handy when testing wireless network security — practically every other tool refers to wireless hosts by their MAC address (or BSSID). The MAC address enables you to easily match up systems you find using NetStumbler, Kismet, or your favorite wireless sniffer with their actual hostnames and IP addresses without having to perform cumbersome reverse-ARP lookups.

Looking for open ports to see what's listening and running on each system is also important. SuperScan is a great tool to use for this because it's easy to use, and it's free! Kevin's partial to SuperScan version 3, as shown in Figure 7-2, because he's been using it for so long, and it simply works.

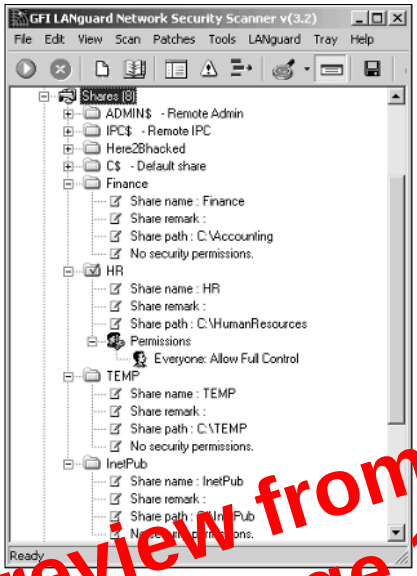

Figure 7-8: Using LANguard Network Security Scanner to find shares on a remote Windows system.

Ferreting Out WEP Keys

Many client vulnerabilities are specific to wireless networks. Standard security tools aren't likely to discover such vulnerabilities. To find these weaknesses, you can use hacking tools that have been created to look for wireless-network vulnerabilities. We discuss such tools below.

Some wireless-specific vulnerabilities require physical access to the computer. It's easy to become complacent and believe that wireless clients are safe because of this physical security requirement, but laptops are lost and stolen quite often, so it's not unreasonable to believe this could occur — especially if users don't report their wireless NICs or laptops stolen. Some vulnerabilities, such as the ORiNOCO WEP key vulnerability, can be exploited by an attacker connecting to the remote computer's registry!

One serious vulnerability affects wireless clients who use the ORiNOCO wireless card. Older versions of the ORiNOCO Client Manager software stores encrypted WEP keys in the Windows registry — even for multiple networks — as shown in Figure 7-9.

You can crack the key by using the Lucent ORiNOCO Registry Encryption/Decryption program found at `www.cqure.net/tools.jsp?id=3`. Make sure that you use the `-d` command line switch and put quotes around the encrypted key, as shown in Figure 7-10. This program comes in handy if you forget your key, but it can also be used against you.

GNUTELLA, GSS-API, HTTP, ICMP, ICQ, IEEE 802.11, IMAP, IP, IRC, ISAKMP, ISDN, KRB5, L2TP, LANMAN, LDAP, LLC,, LSA, LWAPP, LWAPP-CNTL, LWAPP-L3, LWRES, MAPI, NFS, PKCS-1, POP, PPP, PPTP, RPC_NETLOGON, RRAS, RSH, SMB_NETLOGON, SNA, SSH, SSL, Socks, TACACS, TACACS+, TCP, TELNET, TFTP, UDP, VNC, X509AF, X509CE, X509IF, and X509SAT. Fortunately, you can save, print, or filter data.

UNIX/Linux users need the GIMP Toolkit (GTK) for the user interface, whereas the GTK DLLs come bundled with the Windows binary.

You also can use Ethereal as a graphical front-end for packet-capture programs such as Sniffer, `tcpdump`, WinDump, and many other freeware and commercial packet analyzers.

To use Ethereal on a previously created file, you type `tcpdump -r capture.dump` (or WinDump should you wish).

Ethereal is available from `www.ethereal.com`.

This is AirTraf control, you are cleared to sniff

AirTraf was one of the first wireless 802.11b network analyzers. As a wireless sniffer it is a good tool, but does not support wired networks like Ethereal does. It is a passive packet-sniffing tool — it captures and tracks all wireless activity in the coverage area, decodes the frames, and stores the acquired information. AirTraf can record packet count, byte information, related bandwidth, as well as the signal strength of the nodes. You can also run AirTraf in Server Mode, which allows you to have one system that periodically polls other stations to retrieve active wireless data. This is beneficial when you have a large area you want to analyze. You can place AirTraf network analyzers throughout your organization. In this manner, you can consolidate wireless information for your entire organization into a single data store.

AirTraf is Linux open source, and distributed under the GPL license. It is compatible with the 2.4.*x* series of kernels. AirTraf works only with a limited number of wireless adapters. Check the AirTraf Web site to make sure it works with yours.

You can find the freeware AirTraf at `http://airtraf.sourceforge.net/`.

Let me AiroPeek at your data

AiroPeek NX is a Windows-based wireless sniffer that offers some enhanced capabilities, including the ability to detect rogue access, risky device

configurations, Denial-of-Service attacks, Man-in-the-Middle attacks, and intrusions. We have used AiroPeek and highly recommend it. There is one drawback to AiroPeek: It is a commercial product. However, after you use it we think you'll agree that it is money well spent. This is one tool we would recommend that you spend your hard-earned money on if you're going to do more than one ethical hack.

AiroPeek NX comes with a Security Audit Template that creates a capture window and then triggers a notification when any packet matches a specifically designed security filter. This allows the administrator to search for applications like Telnet and access points that use default — therefore *not secure* — configurations.

If you are using Network Authentication with protocols such as Telnet and FTP, you can use AiroPeek to look for failed authentications. These failures might represent an attempted access by an unauthorized person. Once you start to look at the data you are collecting, you can dream up all sorts of similar tests using a sniffer or packet analyzer.

You can find AiroPeek NX at `www.wildpackets.com`.

Another CommView of your data

Another wireless sniffer is CommView for WiFi, which is specific to wireless networks and offers many capabilities besides packet sniffing, such as statistical analysis. By doing statistical analysis, you might find a pattern of unauthorized usage. CommView allows you to grab frames, store the information, and analyze it. CommView for WiFi is a commercial product. You'll find it's not as expensive as AiroPeek but (obviously) more costly than the free Gulpit and Ethereal programs.

When CommView for WiFi is running on your machine, it places your wireless adaptor in passive mode. Your wireless interface can only capture all the packets when it is in passive mode. You will find the installation fairly straightforward since it uses the Windows installer process. Once you install it, you will find many options as shown in Figure 8-1.

You can find Tamosoft CommView for WiFi at `www.tamos.com/products/commview/`.

You cannot obtain data from an access point using WEP or WPA unless you have the appropriate key. You can add key information to CommView for WiFi by selecting Settings➪WEP/WPA Keys and then entering the keys in the areas provided.

Should you use Cain & Abel, be *very* careful. First, you should understand that when using a password cracker, you may violate any number of wiretapping laws or put your organization in a precarious position. If you know passwords are weak and you don't immediately change them, you might have difficulty proving due diligence in a court of law. So ensure that you are on the right side of the law before you touch a key. Second, there is the remote possibility that you could cause damage or the loss of data when using this software or similar tools. These tools intercept packets and may damage these packets. Ensure that you know how the tool works and what it could do — and that good recent backups of system data exist.

The latest version is faster and contains a lot of new features like APR (ARP Poison Routing) that facilitates the sniffing of switched LANs and Man-in-the-Middle attacks. You can use Cain to analyze encrypted protocols such as SSH-1 and HTTPS and to capture credentials from a wide range of authentication mechanisms. It also provides routing protocols authentication monitors and route extractors, dictionary and brute-force crackers for all common hashing algorithms and for several specific authentications, password/hash calculators, cryptanalysis attacks, password decoders, and some not so common utilities related to network and system security. This is indeed the Swiss Army knife for password crackers. Figure 8-3 shows the main window of Cain.

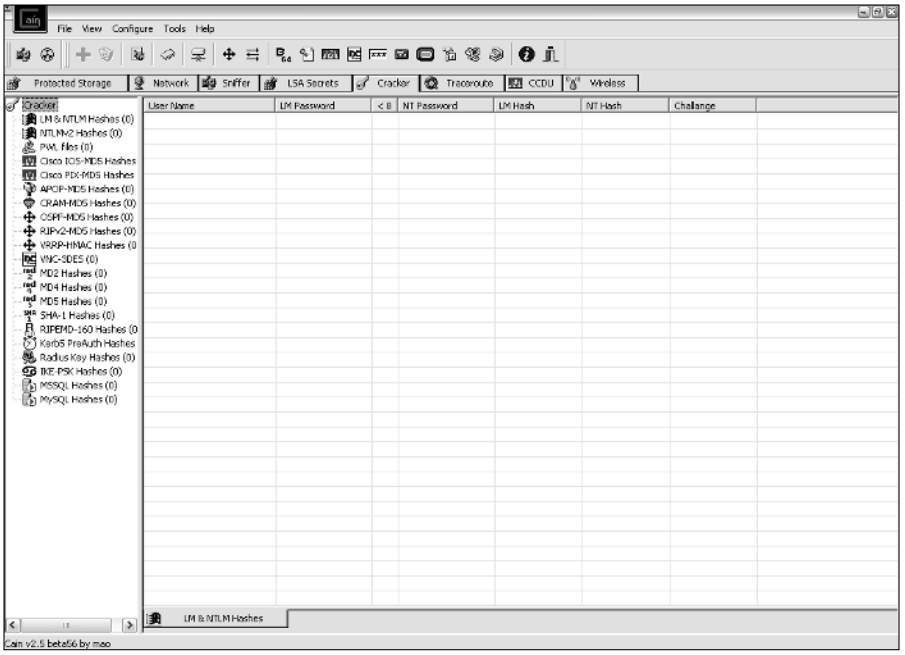

Figure 8-3: Cain main window.

FTP	IRC
Telnet	AIM
HTTP	CVS
POP	ICQ
NNTP	Napster
IMAP	Citrix ICA
SNMP	Symantec pcAnywhere
LDAP	NAI Sniffer
Rlogin	Microsoft SMB
NFS	Oracle SQL*Net
SOCKS	
X11	

`dsniff` benefits the user because it minimally parses each application protocol, saving only the "interesting" data. This speeds up processing.

`dsniff` is really easy to use. Just start it, and it starts listening on the interface you select for passwords.

Mailsnarf outputs all messages sniffed from SMTP traffic in Berkeley mbox format, suitable for offline browsing with a mail reader, such as `pine`. Urlsnarf outputs all requested URLs sniffed from HTTP traffic in Common Log Format, used by almost all Web servers, suitable for offline post-processing with a Web log-analysis tool, such as analog or `wwwstat`. Webspy sends URLs sniffed from a client to a Netscape browser. Filesnarf outputs NFS, SMB, and AFS. Msgsnarf outputs ICQ, AIM, and IRC.

As well, you can use `dsniff` to perform a monkey-in-the-middle attack using `sshmitm` and `webmitm` to sniff HTTPS and SSH traffic and to capture login information.

You can find `dsniff` at `www.monkey.org/~dugsong/dsniff/`. A Windows port is available from `www.datanerds.net/~mike/dsniff.html`, and a MacOS X port is available at `http://blafasel.org/~floh/ports/dsniff-2.3.osx.tgz`.

Gathering IP Addresses

Crackers want targets, and IP addresses are targets. Also, if the wireless administrator is using MAC filtering, then you'll need to gather some IP addresses. You can `ping` every host on a subnet to get a list of MAC to IP

Using SSIDsniff

SSIDsniff is a curses-based tool that allows an intruder to identify, classify, and data-capture wireless networks. The SSIDsniff interface will look familiar if you've ever used the UNIX top utility.

Currently it works under Linux and is distributed under the GPL license. You will need `libpcap` and `curses` or `ncurses` as well. SSIDsniff supports Cisco Aironet and some Prism2 cards.

You can find SSIDsniff at `www.bastard.net/~kos/wifi/ssidsniff-0.40.tar.gz`.

Default-Setting Countermeasures

Okay, even though this chapter introduces you to some very powerful tools, you must not put your head in the sand; just knowing about these tools (and what hackers can do with them) won't make them go away. They are here to stay — and their friends are moving in. Two things we know for sure from the short history of the Internet: These (and other, more insidious tools) proliferate, and they come at you at an ever-increasing pace. Your plan of defense must include ferreting out and trying these tools — as well as their next-generation kid brothers — from here on in. It's an arms race — you must know what the enemy is using, and be prepared to escalate.

The good news is: Some of the countermeasures are decidedly low-tech. There's really no excuse for not implementing them.

Change SSIDs

When you get a new system, you must ensure that you change the default SSID. We know Linksys uses `Linksys` as a default SSID (obvious, much?), and we know others as well. When picking a new SSID — as long as we're talking obvious (but vital) here — don't select one that's easy to guess. Even though the SSID is most emphatically not a password, there is no reason to select an easy-to-guess one.

If you don't know what the default SSID is for a particular access point, you can find it out at one of the following Web sites:

- `www.cirt.net/cgi-bin/passwd.pl`
- `www.phenoelit.de/dpl/dpl.html`
- `http://new.remote-exploit.org/index.php/Wlan_defaults`
- `www.thetechfirm.com/wireless/ssids.htm`

Don't broadcast SSIDs

In this chapter, we showed you that even when you don't broadcast your SSID, others can derive it. But that doesn't mean you shouldn't disable it. When someone roams your neighborhood running NetStumbler, make it more difficult for them. Disable your SSID broadcasting and make them come back running Kismet. You may not have defeated them (yet), but you've at least made things more difficult for them.

Using pong

Older readers probably think pong is a video game. If you are a computer virus researcher or fighter, then you probably think pong is a nasty Trojan. Well, this pong is neither, but rather a tool to check the vulnerability of your wireless access point. If your access point is running vulnerable firmware, pong will give you access to all relevant details such as the admin password, WEP keys, allowed MAC addresses, and more. Should pong work successfully against your network, then you'll need to upgrade your firmware to protect yourself.

Pong is a DOS program and is easy to use, just type `c:\> pong [-r]` in a command shell. The `-r` option provides additional raw output of all received data. When pong finds an access point from the following list, you will get a list of all relevant parameters:

- 4MBO
- Airstation
- D-Link DWL-900AP+
- Linksys
- Melco
- US Robotics
- Wisecom

You can find pong at `http://mobileaccess.de/wlan/index.html?go=technik&sid=`. Praemonitus, praemunitus. (Or for those of you who don't still speak Latin, that's *forewarned, forearmed.*)

Detecting sniffers

At Layer 2, you can run LBL's arpwatch (`www.securityfocus.com/tools/142`) to detect changes in ARP mappings on the local network, such as those caused by `arpspoof` or `macof`.

At Layer 3, you can use a tool such as AiroPeek, CommView for WiFi, or any other programmable sniffer (say, NFR) to look for either the obvious network anomalies or the second-order effects of some of `dsniff`'s active attacks. If you want to learn how to use a packet analyzer for security, try one of Laura Chappell's network analysis or troubleshooting books that you can download for a fee from `www.packet-level.com/books.htm`.

Also, anti-sniffing programs such as l0pht's AntiSniff (`http://packetstormsecurity.nl/sniffers/antisniff/`) can uncover `dsniff`'s passive monitoring tools.

Chapter 9

Wardriving

In This Chapter

- Installing and configuring Network Stumbler
- Running NetStumbler
- Interpreting the results
- Mapping and viewing the results

When most people think of wireless security (or the lack of it), they think of someone driving around their neighborhood discovering their access point and trying to connect. This is a striking image: A nerd in a car by himself with his beloved laptop and some arcane software. It's an activity called *wardriving,* and though it seems hostile at first blush, the reality is actually a lot more diverse. In effect, wardriving is an educational opportunity for everyone — especially for ethical hackers. Peter, for example, actually goes wardriving with his teenage daughter. After all, the family that drives together, strives together.

In this chapter, we take our first look at wardriving. To understand this genre of software, we will look at Network Stumbler (a.k.a. NetStumbler). We'll also see how to map the results of your work. In Chapter 10, we discuss other examples of wardriving software, such as Kismet and Wellenreiter.

Introducing Wardriving

The term *wardriving* is derived from the phrase *war dialing.* But it really doesn't involve guns or offensive weapons of any kind. Wardriving is just the term coined for wireless network discovery. Nothing more or less. In Chapter 4, we outlined the tools you need for your wardrive, but all you need to wardrive is some software and a wireless network interface card or adapter. If you really want to get into it, you can add an external antenna to enhance the signal strength of any access points that you find. This enables you to detect these access points at a greater distance than when you were only using the built-in antenna of your wireless NIC alone. You could also add a global positioning system (GPS) to map the latitude and longitude of the networks you find.

Setting Up NetStumbler

After NetStumbler starts, you may want to set the options to maximize your wardriving experience.

Figure 9-1 shows data from an actual wardriving session, shot after the session. Looking at the window, you can see a left and a right pane. The status bar beneath the panes provides some valuable information. The message in the middle of the status bar tells you how many access points are active. To the right of that is the status information. You can find descriptions of the possible status messages in Table 9-1. The last piece of information on the bar right tells you how many networks NetStumbler found. In our case, it found 461. The number before the slash tells you how many networks meet the criteria or filter that you selected from the left pane. If you are looking at the main screen and not filtering anything, then the first and second number are the same. Anytime you select anything from the left-hand pane, the first number will change. For example, when I select the Encryption Off under Filters, the number is 253 of 461, or about 55 percent of my neighbor's networks don't use encryption. (You can get a closer look at the two panes later in the chapter, after we talk about the setup options.)

Table 9-1 Status Message

Message	*Description*
Card not present	Wi-Fi card not detected. Make sure you have installed a wireless NIC.
A device attached to the system is not functioning	Problem working with the Wi-Fi card. Switch interface mode on the device menu.
Not scanning	Scanning is not enabled. Click the arrow or start from the File menu.
No APs active	Wi-Fi card is working, but not detecting any networks at the time.
x APs active	Wi-Fi card is working and detecting x number of networks.
GPS: Acquiring	NetStumbler is receiving a message from the GPS.
GPS: Disabled	The GPS is disabled. Start it to record network coordinates.
GPS: Disconnected	The GPS was working but stopped. Check the GPS power.

Table 9-4 lists the parameters, describes them, and provides the options or settings you may choose.

Table 9-4 GPS Options

Option	*Description*	*Settings*
Protocol	Format of the GPS data	NMEA 0183, Earthmate, Garmin Binary, Garmin Text, or Tripmate
Bits per second	Transfer rate from the GPS	110 to 256000
Data bits	Number of bits used for data	5 to 8
Parity	Parity bits	Mark, One, Odd, or Space
Port	Communication port for the GPS	Disabled or COM1 to COM16
Stop bits	Number of bits used for communication	1, 1.5, or 2
Flow control	Handshaking protocol	None, Hardware, or Xon/Xoff

The NMEA standard sends a signal to NetStumbler every 2 seconds, whereas the Garmin standard sends it once per second.

Check the manual that comes with your GPS; it should tell you the settings you need.

Selecting Scripting options

NetStumbler lets you modify its operation through the use of scripts. You may choose to use common scripting languages such as PerlScript, Python, VBScript, Jscript, Windows Script Components, Windows Script Host, and Windows Script Runtime version. After you write your script, install it on the same system as Network Stumbler and then make it known by clicking the Scripting tab of the Network Stumbler Options dialog box. Do so and you should see the options shown in Figure 9-6.

Select the Type, File name, scripting Language, and Status of the script. Then when NetStumbler starts, it will execute the script. You can find a scripting guide at `www.stumbler.net/scripting.html`. Also, you might want to check out the Scripts Forum at

```
http://forums.netstumbler.com/forumdisplay.php?s=&forumid=24
```

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

Option	*Description*
`-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 gpsd to open a listener. This is not the "listening port" for the gps 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.

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.)

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 /) is network discovery done from a motorcycle or bicycle.
- *Warflying* (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.jiffie.com/warchalking-introduction.html 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 hobos 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) 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] 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.

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`) 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 among the horde of fake beacon frames. As part of a honeypot or flying solo, Fake AP confuses NetStumbler 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 neighbor's AP, 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`), 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/`).

Chapter 12

Network Attacks

In This Chapter

- Understanding the consequences of attacks on wireless systems at the network level
- Unmasking MAC address spoofing
- Unmanning man-in-the-middle attacks
- Reviewing known problems with SNMP
- Defining the Queensland protocol attack
- Examining the quirky network issues with network analyzers
- Exploring practical and cost-effective countermeasures

Your computer systems and applications require one of the most fundamental communications systems in your organization — your network. Although many organizations don't completely rely on wireless networks for everything, others do. Either way, your wireless network likely depends on critical servers; you can't afford to have them compromised via the network. These computers, even if they're an ancillary part of your overall network, are there for business reasons; damage them, damage the business. Therefore it's important to understand just what can happen when network-based 802.11 vulnerabilities are exploited.

There are thousands of possible network-level vulnerabilities on your wireless systems — and seemingly just as many tools and testing techniques. The key point to remember here is that you don't need to test your wireless network for *every* possible vulnerability, using every tool available and technique imaginable. Instead, look for vulnerabilities that can have a swift and immediate impact on your systems.

Some of the hacks and associated tests we demonstrate in this chapter are specific to 802.11. Others are security weaknesses common to any network — and those not only have a higher likelihood of being exploited, they can also have a high impact on your business.

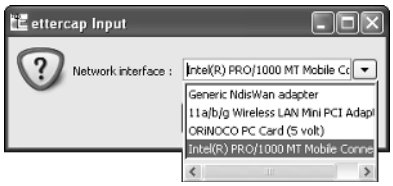

Figure 12-12: Selecting a NIC for Ettercap NG to use.

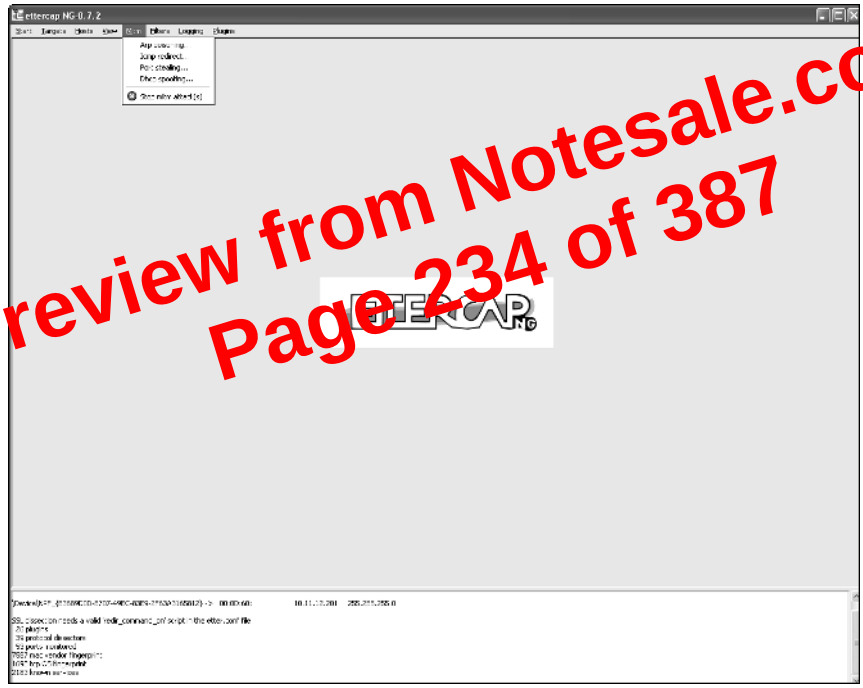

Figure 12-13: Selecting the MITM attack of your choice in Ettercap NG.

Again, note how simple it is to achieve a MITM attack. At this point, you can use Ettercap NG and your favorite network analyzer to capture your victim system's data — or launch other attacks of the type mentioned in this section.

SNMP: That's Why They Call It Simple

Simple Network Management Protocol (SNMP) is a protocol built in to virtually every network infrastructure device — both wireless and wired. Everything from switches to routers to servers to APs can be managed via SNMP. There

All Hail the Queensland Attack

A relatively new attack against the 802.11 protocol showed up Down Under in May 2004, discovered by researchers at Queensland University of Technology's Information Security Research Centre (`www.kb.cert.org/vuls/id/106678`) in Australia. This attack, initially referred to as the Clear Channel Assessment attack, affects the Direct Sequence Spread Spectrum function that works as part of 802.11's Carrier-Sense Multiple Access/Collision Avoidance (CSMA/CA) protocol that manages the wireless communications medium. This attack is often called the *Queensland Attack* — crediting the researchers who discovered it.

Wireless systems (clients, APs, and so on) use CSMA/CA to determine whether or not the wireless medium is ready and the system can transmit data. The Queensland attack exploits the Clear Channel Assessment (CCA) function within CSMA/CA and basically makes it appear that the airwaves are busy — effectively preventing any other wireless system from transmitting. This denial of service is accomplished by sending a wireless NIC in continuous transmit mode.

With the right tool, the Queensland Attack is relatively simple to execute. It can wreak havoc on a wireless network, effectively bringing it to its knees. There's very little that can be done about it, especially if the attacker's signal is more powerful than that of your wireless systems. That's no problem for hackers equipped with a high-powered wireless NIC combined with a high-gain antenna (see Chapter 13 for more information). Combine an easily overpowered network with the fact that 802.11 systems use a shared medium to communicate, and you have the makings of a very effective attack.

All it takes for an attacker to run such an attack against your wireless systems is to run an old Prism chipset-testing program called Prism Test Utility (`PrismTestUtil322.exe`). This program was previously available for public download on Intersil's Web site — and it's still easy to find elsewhere with a basic Internet search, so it's probably not going away any time soon. This attack can just as easily be carried with other hardware tweaking as well.

Although the Queensland Attack exploits an 802.11 protocol issue, it could just as easily be considered a DoS attack, given its outcome (big-time denial of service). Refer to Chapter 13 for an in-depth look at various wireless DoS attacks.

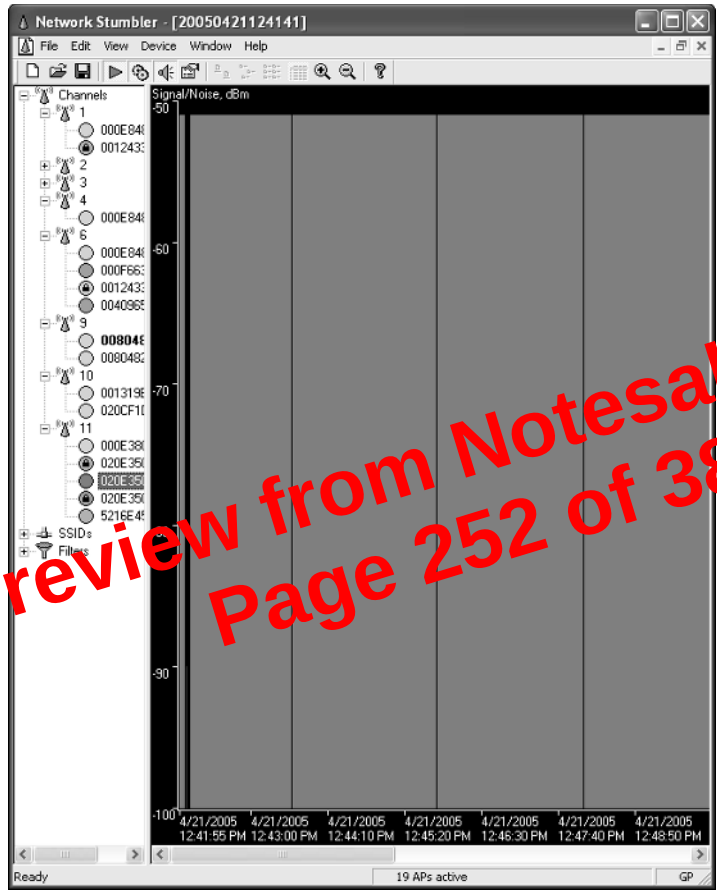

Figure 13-1: Strong wireless signal experiencing no interference.

Figure 13-2 shows an 802.11b signal that's experiencing some random noise and signal loss. Notice the signal profile: It's degraded and choppy compared to that of Figure 13-1.

Figure 13-3 shows an 802.11b signal that's experiencing severe jamming. Notice that although the signal is strong at times, it's missing across various time periods and is being overpowered by another signal. This secondary signal is shown in red at the bottom of the green (actual) signal in NetStumbler. NetStumbler also shows a purple bar that signifies a potential loss of radio signal.

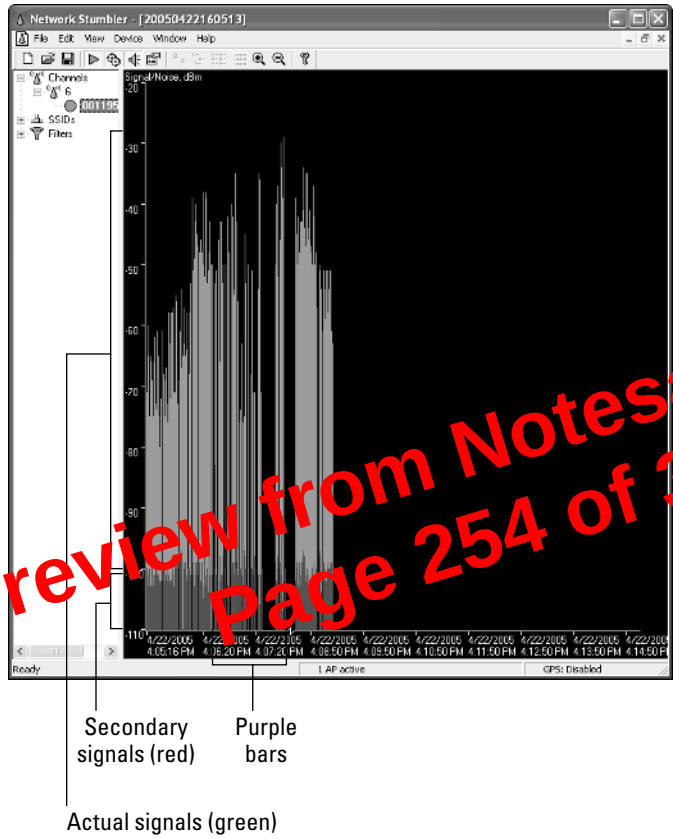

Figure 13-3: Wireless signal experiencing severe signal disruption (jamming).

There are also various signal-generator vendors listed at online e-commerce sites such as Naptech (`www.naptech.com`) and TestMart (`http://signalgenerator.testmart.com`).

A jamming attack against a wireless network can be carried out from several dozen meters away, which helps the attacker hide. The two jammers we mentioned are handheld systems — so an attacker could conceivably have one stored in his pocket or briefcase, and you'd be none the wiser. Perhaps the most frustrating thing about jammers is that even the most highly protected wireless systems are pretty much indefensible in the face of such an attack.

We won't demonstrate what using a radio power generator can do to a wireless network — but suffice it to say that the outcome is likely to be worse than the RF signal disruption shown earlier in Figure 13-3.

AP Overloading

802.11-based wireless access points can only handle so much traffic before their memory fills up and their processors become overloaded. This type of DoS attack overloads not only the wireless medium (as outlined earlier) but also the actual wireless infrastructure — and APs themselves.

There are several ways that APs can become overloaded and simply stop addressing the needs of existing or new clients — or just break down altogether. Some of these de-facto attacks are unintentional; others are deliberate and malicious. Let's take a look at what can happen.

Guilty by association

Attackers can exploit a weakness in the way access points queue incoming client requests — beginning with the *client association identifier (AID) tables* — the section of an AP's memory that stores client connection information. The AID tables only have a finite amount of memory and thus can only handle a limited number of wireless client connections. Once this memory fills up, most APs will no longer accept incoming association requests; some APs even crash.

These types of DoS attacks typically use one of two methods:

- Association flooding
- Authentication flooding

Both are easier to do when anybody can connect. When APs are set up to use "open" as the default authentication type, just about any client (trusted or untrusted) can connect to the AP. This is one of those fundamental 802.11 security flaws deemed necessary to keep wireless-connectivity headaches to a minimum. Such *open authentication* allows any client to send two critical requests:

- Authentication requests for initial connectivity
- Association requests to "join" the wireless network

Now, wireless client connectivity to an AP that's running open authentication has the three basic phases:

1. No connection
2. Authenticated but not associated
3. Authenticated and associated

This three-step process is critical for understanding DoS attacks, so we show it again in Figure 13-4.

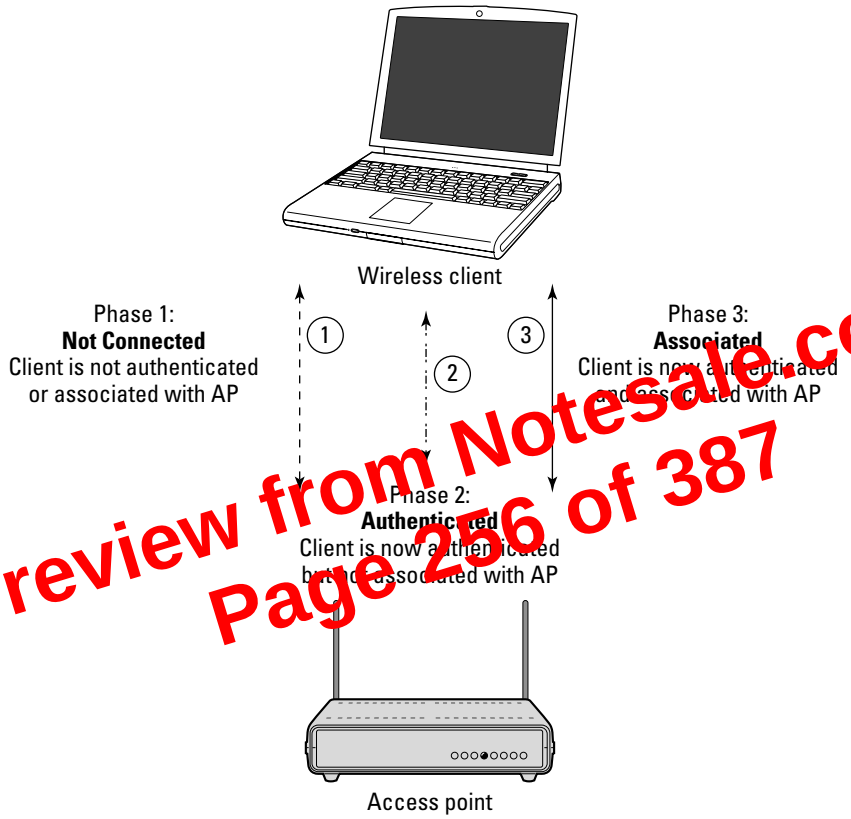

Figure 13-4: Client-to-AP connection process.

Attacks that overload the AID tables create a situation that can take a wireless network from normal to frozen in no time: Even an average number of legitimate wireless-client connections can multiply to an insane number when illegitimate connections pile on, faster than you can say *intrusion prevention.*

Association and authentication attacks are possible mainly because 802.11 management-frame requests and sequencing are not authenticated — or monitored for anomalies.

If you're up for testing to see how easy it is to fill up the AID tables on your AP(s), there are several tools you can use. One of our favorites is Void11 — a packet-injection tool. Figure 13-5 shows its options: Notice the authentication- and association-flood options, as well those for flooding a single target, broadcast systems, and randomly generated systems.

Packet Generator, which is very easy to use, allows you to replay practically any 802.11 packet (including Association and Authentication Request packets) that you've captured in CommView for WiFi or another network-analyzer program.

Here's a brisk walkthrough capturing an association request packet in CommView for WiFi, copying the packet to the Packet Generator tool, and then sending the packet onto the airwaves:

1. **Load CommView for WiFi and click the blue Start Capture icon in the upper-left corner or simply press Ctrl+S on your keyboard.**

   This loads the Scanner utility (as shown in Figure 13-7) so you can enable your wireless NIC to capture packets.

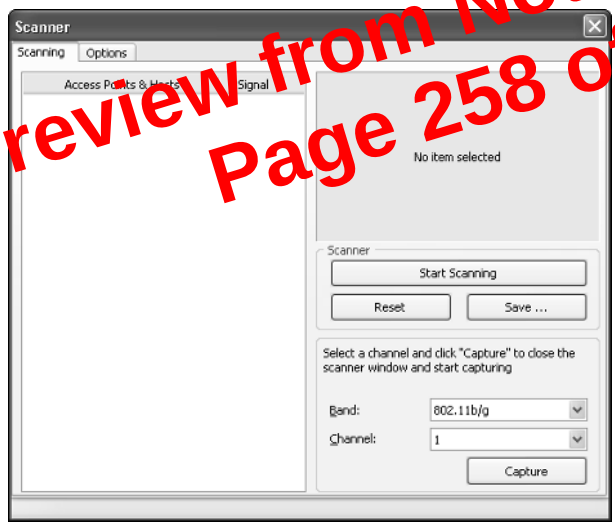

Figure 13-7: CommView for WiFi's Scanner utility.

2. **Click the Capture button on the Scanner window.**

   This "opens" the Wireless Adapter Enable Promiscuous mode on your wireless NIC, and allows you to start capturing wireless packets.

3. **Capture an Association Request packet.**

   The easiest way to do this is to power on a new wireless client and look for its requests to the AP to associate. Packet number 115 in Figure 13-8 shows what an Association Request packet looks like. Note that CommView for WiFi lists this as a MGNT/ASS REQ. packet where the MGNT represents a *management* type packet.

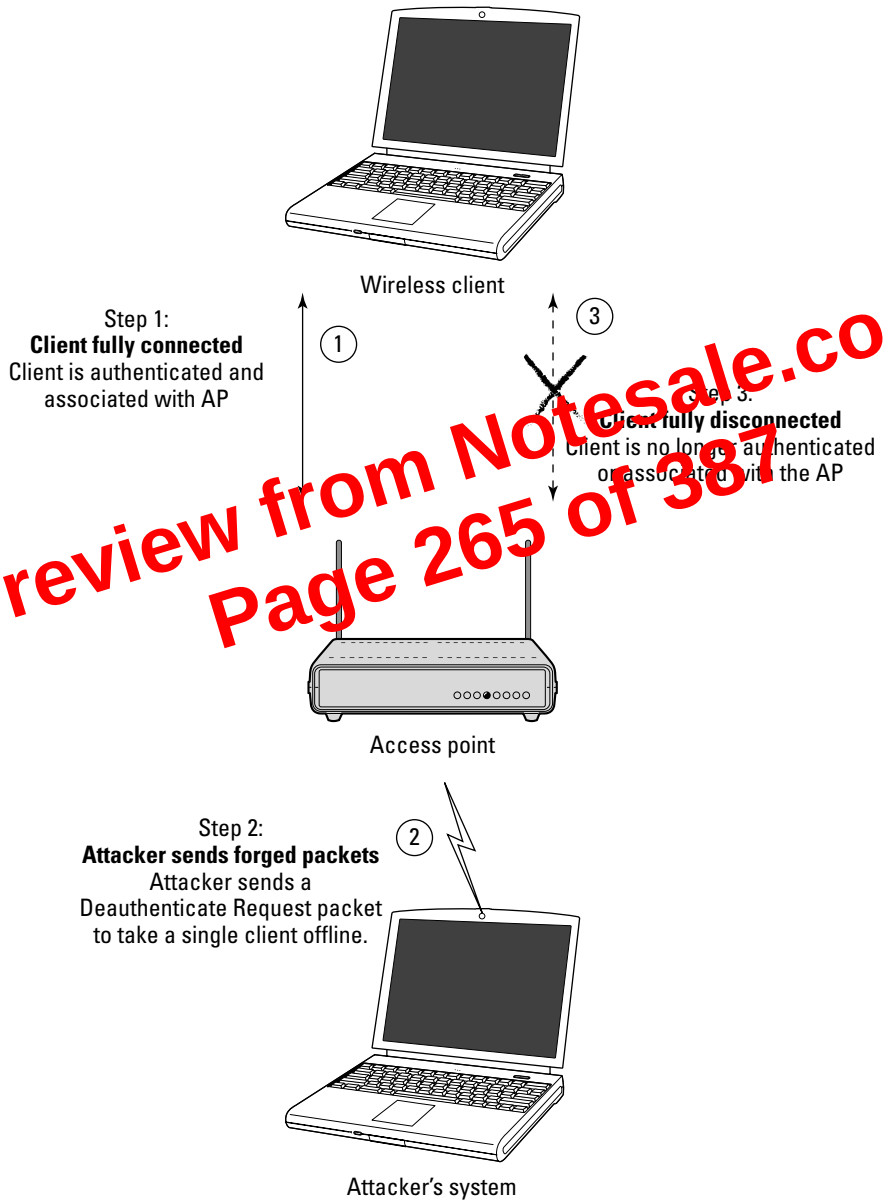

Figure 13-11: Deauthentication attack completely disconnecting a wireless client.

If you care to see how your systems respond to deauthentication attacks, here's how it can be done using CommView for WiFi:

1. **Load CommView for WiFi and click the blue Start Capture icon in the upper-left corner or simply press Ctrl+S on your keyboard.**

   This loads the Scanner utility as shown in Figure 13-7 above so you can enable your wireless NIC to capture packets.

2. **Click the Capture button on the Scanner window.**

   This "opens" the Wireless Adapter Enable Promiscuous mode on your wireless NIC and allows you to start capturing wireless packets.

3. **Generate a Deauthentication packet.**

   It's a little trickier capturing one of these packets. But if you have an AP that supports manual deauthentications, capturing can be pretty simple. As shown in the Cisco management screen in Figure 13-12, it's as easy as clicking the Deauthenticate button for the client you wish to deauthenticate.

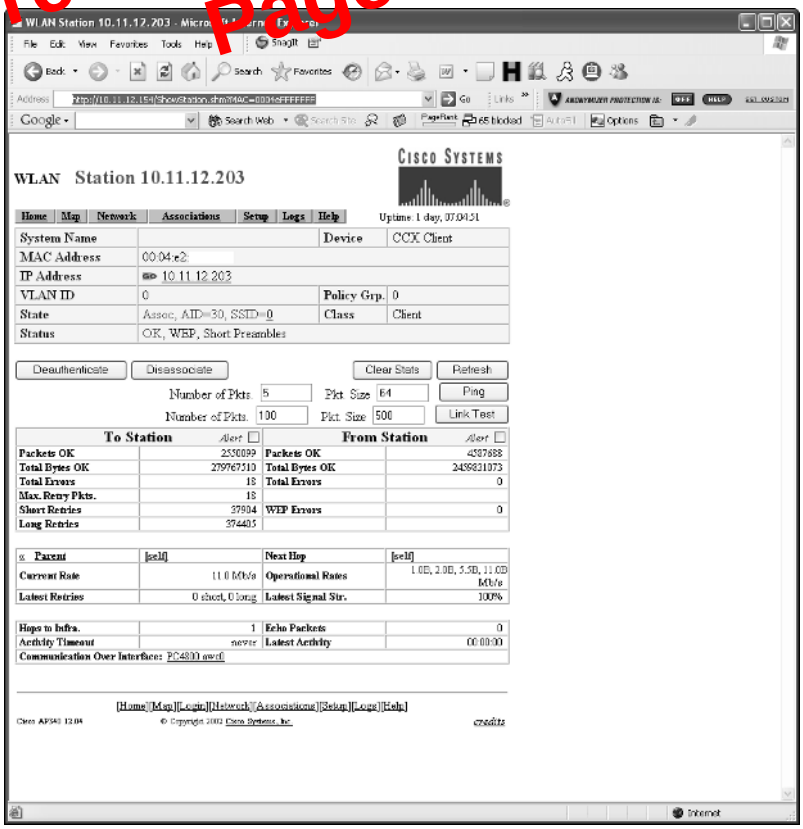

Figure 13-12: Cisco Aironet option to deauthenticate a wireless client.

4. **Capture the Deauthentication packet.**

   This is as simple as capturing all wireless packets — or narrowing it down to management packets — in a network analyzer. Figure 13-13 shows what such a packet looks like in AiroPeek. All you have to do is capture the packet using any wireless network analyzer, save the packet, and import it into CommView for WiFi's Packet Generator. Or you can simply capture the packet in CommView for WiFi and save the packet using the steps we outlined for the Association Request packet above.

5. **Edit the Deauthentication packet.**

   After you have the packet loaded into CommView for WiFi's Packet Generator, you can edit it to change source and destination addresses. In this example, we'll change the source address to effectively turn it into a forged address and change the destination address to the broadcast address.

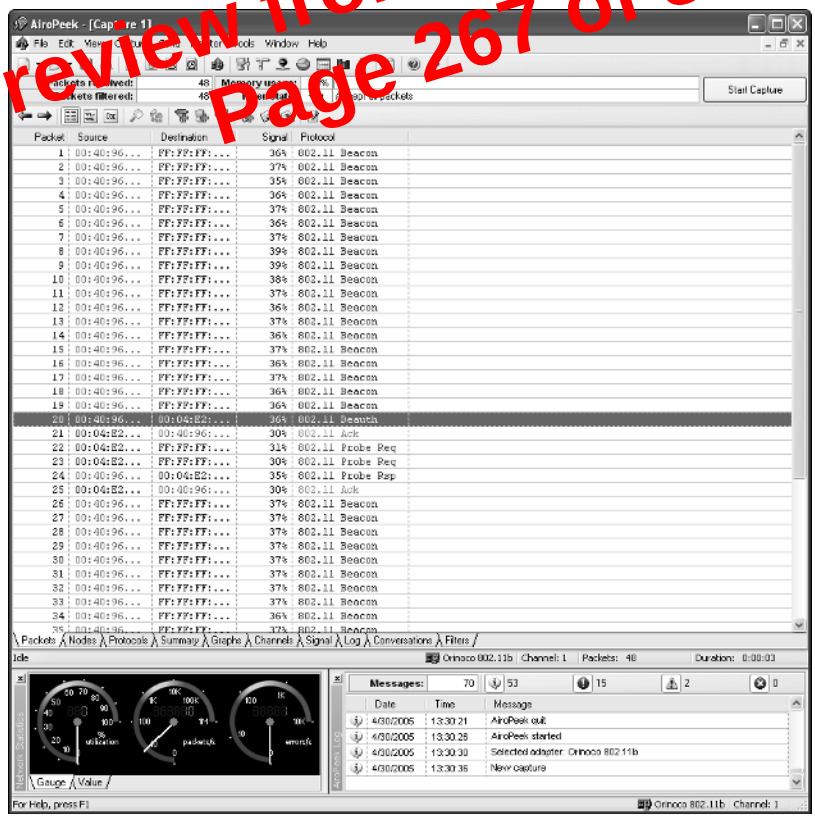

Figure 13-13: A Deauthentication packet discovered by AiroPeek.

For a real-world view of what this type of attack can do to a wireless client, take a gander at Figures 13-17 (normal wireless connectivity and a test ping out to a Web site) and 13-18 (the havoc after deauthentication).

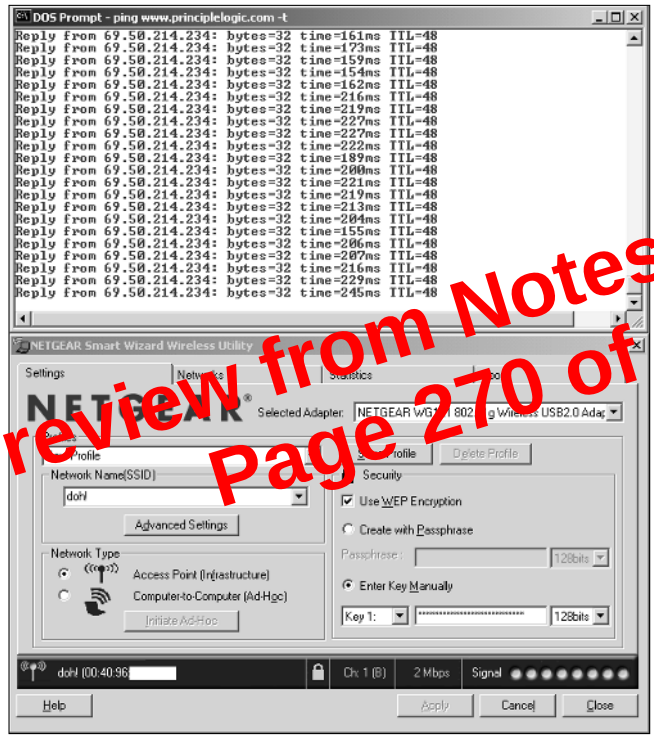

Figure 13-17: Normal wireless client connectivity.

Invalid authentications via fata_jack

There are other tools that can create similar client DoS attacks. One popular one is Mark "Fat Bloke" Osborne's `fata_jack`. This is a Linux program based on the `wlan_jack` program that you'll have to compile before using. It sends out invalid Authentication Failed frames, allowing an attacker to spoof a valid client on the network and send these invalid frames to the AP. The AP, in effect, responds to the client with *Hey! Your previous authentication failed, so forget you — I don't want to speak to you any more.*

This attack is known to create erratic behavior on wireless clients, especially those running on older operating systems with older wireless hardware. Before using this program, you compile it (via the instructions in the source code); then you can run it to see whether any of your systems are vulnerable — just be careful so you don't crash critical systems.

Chapter 14

Cracking Encryption

In This Chapter

- Understanding encryption
- Encrypting frames
- Looking at WEP problems
- Upgrading to WPA
- Using AES
- Tunneling through the Internet with a VPN

Most people believe that encryption is a panacea. They believe that when you encrypt something, it's secure. Unfortunately, this is just not true. As with many newer technologies, you may find the available security features of encryption not as comprehensive or robust as you might like. Cryptography features can have flaws. You can use the wrong algorithm, a flawed algorithm, a short key, or a poor implementation, and (oops!) there it is: a security breach. This chapter demonstrates how one or more of these problems affects the use of encryption with your wireless networks.

But we don't want to play Cassandra and bring only bad news. We also show you some techniques for strengthening your access point. At a minimum, we strongly recommend that you use the built-in security features as part of an overall defense in-depth strategy.

What Can Happen

The IEEE 802.11 specification identified features that a wireless network needs to maintain a secure operating environment. One of the primary features was the use of encryption to provide the following:

- **Message privacy:** Sensitive information is encrypted when transmitted between two wireless entities to prevent interception and disclosure or prevent a third party from tracking communications between two other entities.

the exception of a fully switched environment, eavesdroppers can have their way with frames traversing a wired network. WEP was never intended to provide message integrity, non-repudiation, and confidentiality. And guess what — it doesn't.

WEP uses the symmetrical RC4 (Ron's Code 4) algorithm and a PRNG (Pseudo-Random Number Generator). The original standard specified 40 (in practice, 64) and 128-bit key lengths with a 24-bit initialization vector (IV). Then there's the matter of incomplete coverage of network layers: WEP encrypts Layers 3 through 7, but does not encrypt the MAC layer (that is, Layer 2). Because it's a symmetrical algorithm, WEP gives every client the keys and other configuration data.

Okay, we know there's nothing wrong with the RC4 algorithm per se — after all, Web browsers use it for Secure Socket Layer (SSL). The problem is in the WEP implementation of the RC4 algorithm — and the false sense of security it encourages.

The algorithm takes the IV, which is in plaintext, and sticks it on the front end of the secret key (which the decrypter knows). WEP then plugs the result into the RC4 to receive a key stream. Next, the algorithm XORs the key stream with the ciphertext, which should give us the plaintext value. Finally, WEP re-performs the CRC-32 checksum on the message and ensures that it matches the integrity check value in our encrypted plaintext. Should the checksums not match, WEP assumes that someone tampered with the packet, and will discard it.

As mentioned earlier, access points generally have only three (namely, the following) encryption settings available:

- **None:** This setting represents the most serious risk because someone can easily intercept, read, and alter unencrypted data traversing the network.
- **40-bit shared key:** A 40-bit shared key encrypts the network communications data, but there is still a risk of compromise. The 40-bit encryption has been broken by brute force cryptanalysis, using a high-end graphics computer — and even low-end computers — so it has only questionable value. We show you some tools in later sections that allow you to easily recover 40-bit keys — and if you can, a bad guy can.
- **104-bit setting:** In general, 104-bit (sometimes called 128-bit) encryption is more secure than 40-bit encryption because of the significant difference in the size of the cryptographic key space. Even though this better security isn't true for 802.11 WEP (because of poor cryptographic design in the use of IVs), it is nonetheless recommended as a good practice. Again, you should be vigilant about checking with the vendor regarding upgrades to firmware and software — you may find some that overcome some of the WEP problems. (Some vendors, for example, support 152-bit keys.)

The manufacturer may provide one or more keys to enable shared-key authentication between the device that's trying to gain access to the network and the AP. And yes, we're going to say it again: Using a default shared-key setting is a security vulnerability — a common one because many vendors use identical shared keys in their factory settings. A malicious cracker may know the default shared key and use it to gain access to the network.

Don't use default WEP keys! No matter what your security level, your organization should change the shared key from its default setting because it's just too easily exploited. In the event you don't know the default keys for a wireless access point (or you don't know whether there is a default key), check out `www.cirt.net`.

Some products generate keys after a keystroke from a user. If done properly using the appropriate random processes, can result in a strong WEP key. Other vendors, though, based WEP keys on passwords chosen by users; this typically reduces the effective key size.

You may find your configuration utility doesn't have a passcode generator, but allows you to enter the key as alphanumeric characters (that is, a to z, A to Z, and 0 to 9) rather than as a hexadecimal number. You just need to create a good passcode, right? Sounds like a good idea — until you study it. Each character you enter represents 8 bits, so you can type 5 characters for a 40-bit code and 13 characters for a 104-bit code. Entering 5 characters in ASCII is not as strong as generating the key randomly in hexadecimal. Think of all the poor five-letter passcodes you could create!

So take it from us: WEP is weak. The following is a summary of some of the more glaring weaknesses of WEP:

- The IV value is too short — and not protected against reuse.
- The way keys are constructed from the IV makes it susceptible to weak key attacks.
- There is no effective detection of message tampering; that is, WEP has no effective message integrity.
- It directly uses the master key and has no built-in provision to update the keys.
- There is no provision against message replay.
- There is no key-management mechanism built in.

At a minimum, enterprises should employ the built-in WEP encryption. But that's a poor minimum. And it's amazing how many access points don't have any encryption at all. We find that less than half the access points we stumble on have encryption of any sort.

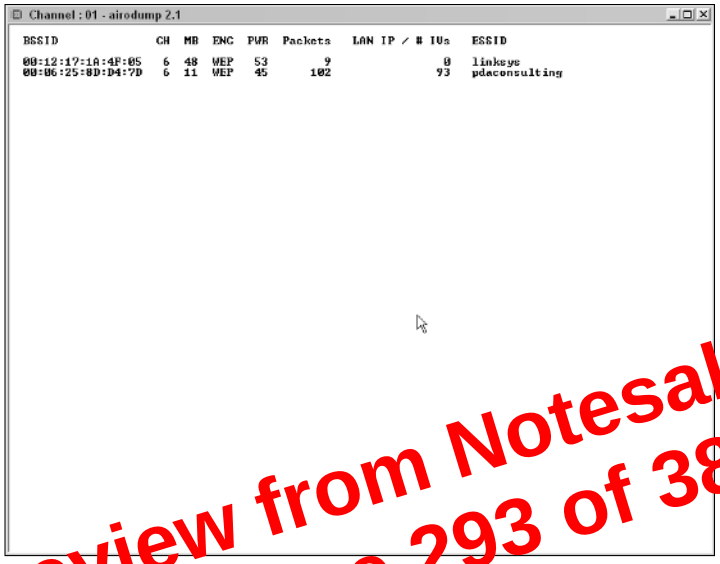

Figure 14-8: Airodump capturing frames.

Aircrack often determines a WEP key within a few seconds, but the execution time is highly variable. It took Peter several days to crack a WEP with little activity and a 40-bit key. Shorter execution times require more traffic, more unique IVs, more luck, and the lowest successful *fudge factor,* a setting that tells aircrack how wildly it should guess when trying new keys. The higher the fudge factor, the more keys aircrack will try — increasing both the potential time of execution and the likelihood that the attack will succeed. The fudge factor has a default value of two, but you can set it to any positive integer. The default setting is a good place to start, but try several different settings when the initial attack does not succeed. Note, however, that there's a tradeoff: Generally the higher the fudge factor, the longer the execution time.

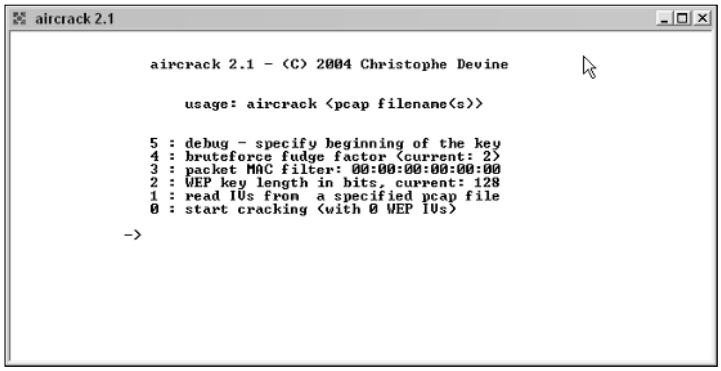

Figure 14-9: Aircrack window: options.

Only time will tell whether there are flaws in 802.11i. We cover 802.11i features in this section — and again, later in the chapter, when we cover AES. Implementing 802.11i requires new hardware. Not everyone wants (or needs) to acquire new hardware — but everybody will still want improved security. So far, it looks as if WPA comes to the rescue.

An initiative for improving WLAN security is the interim solution — Wi-Fi Protected Access (WPA) — to address the problems of WEP. WPA uses the Temporal Key Integrity Protocol (TKIP) to address the problems without requiring hardware changes — that is, requiring only changes to firmware and software drivers. TKIP is also part of the RSN.

WPA is an example of a software or firmware patch. As an interim security solution, WPA does not require a hardware upgrade to your existing 802.11 equipment; the full-blown 802.11i standard does. WPA is not a perfect solution, but it does attempt quick and proactive delivery of enhanced protection to address some of the chronic WEP problems that predate the availability of 802.11i security features. WPA has two key features:

- 802.1X support
- Temporal Key Integrity Protocol (TKIP)

WPA uses 802.1X port-access control to distribute per-session keys. (Some vendors previously offered 802.11X support, even though it wasn't specified in the standard.) The 802.1X port-based access control provides a framework that allows use of robust upper-layer authentication protocols.

Temporal Key Integrity Protocol (TKIP) provides key mixing and a longer initialization vector. It also provides a Message Integrity Check (MIC) that prevents wireless data from being modified in transit. Even better, TKIP offers some essential support for wireless-network security by

- Managing keys to prevent the reuse of a static key
- Facilitating the use of session keys because cryptographic keys should change often
- Including four new algorithms to enhance the security of 802.11
- Extending the IV space
- Allowing for per-packet key construction
- Providing cryptographic integrity
- Providing key derivation and distribution

Through 802.11i and WPA, TKIP protects against various security attacks discussed earlier in this chapter — including replay attacks and attacks on data integrity. Additionally, it addresses the critical need to change keys. Again, the objective of WPA was to bring a standards-based security solution to the

First off, it's worth looking at the three states a wireless client goes through in the authentication process:

- **Unauthenticated and unassociated:** The client selects a basic service set by sending a probe request to an access point with a matching SSID.
- **Authenticated and unassociated:** The client and the access point perform authentication by exchanging several management frames. Once authenticated, the client moves into this state.
- **Authenticated and associated:** Client must send an association request frame, and the access point must respond with an association response frame.

A client can authenticate to many access points, but will associate only with the access point with the strongest signal.

In the second state, we just casually mention the client authenticates to the access point. It's not quite that simple.

Authentication according to IEEE 802.11

The IEEE 802.11a and b specifications define two ways to "validate" wireless users who are attempting to gain access to a wired network. One does the job; the other one doesn't:

Open-system authentication: convenient but dangerous

This "authentication" technique isn't really authentication because the access point accepts the mobile station willy-nilly without verifying its identity. The access point authenticates a client when the client simply responds with a MAC address during the two-message exchange, in a simple (and insecure) process:

1. Client makes a request to associate to an access point.
2. The AP authenticates client and sends a positive response — voilà! The client is associated.

Shared-key authentication

Shared-key is a cryptographic — that is, real — technique for authentication. It is a simple "challenge-response" scheme based on whether a client has knowledge of a shared secret. In this scheme, the access point generates a random 128-bit challenge that it sends to the wireless client. The client, using a cryptographic key that is shared with the access point, encrypts

Uh-oh. And if you're with us so far, then the rest is just simple math:

```
If P XOR R = C then C XOR R = P
If P XOR R = C then C XOR P = R
```

where P = plaintext, C = ciphertext, and R = key stream (or random bytes). Now the attacker knows everything: algorithm number, sequence number, status code, element ID, length, and challenge text. It's the attacker's turn, and here's how it looks, blow by blow:

1. The attacker requests authentication.
2. The access point responds with a cleartext challenge.
3. The attacker uses the challenge with the value R (as just shown) to compute a valid authentication-response frame by XORing the two values together. Result: He can compute a valid ICRC value.
4. The attacker responds with a valid authentication-response message and associates with the AP to join the network.

The attacker did not need to know the shared-key due to the flaw! (Welcome aboard, stranger. Oops.)

Due to the problems with shared-secret authentication, the standard developers specified WPA and WPA2, both using 802.1X with Extensible Authentication Protocol (EAP).

Have We Got EAP?

So what *is* 802.1X? Did we mean 802.11*x*? No, 802.1X is another IEEE standard, which provides a framework for true user authentication and centralized security management. It provides port level authentication. Initially, the developers offered to standardize security on wired network ports, but others found that the standard had applicability for wireless networking as well.

EAP (Extensible Authentication Protocol) has three components:

- **The supplicant:** A client machine trying to access the wireless LAN.
- **The authenticator:** A Layer 2 device that provides the physical port to the network (such as an access point or a switch).
- **The authentication server:** This verifies user credentials and provides key management.

You can find THC-LEAPcracker at `http://thc.org/releases.php?s=4&q=&o=`.

Using anwrap

Written by Brian Barto and Ron Sweeney, `anwrap` is a wrapper for `ancontrol` that serves as a dictionary-attack tool against LEAP enabled Cisco Wireless Networks. It traverses a user list and password list, attempting authentication and logging the results to a file. `anwrap` causes havoc on NT Networks that have lockout policies in place.

`anwrap` requires ancontrol and Perl. The `ancontrol` command controls the operation of Aironet wireless networking devices via the `an` driver. The `anwrap` author tested the tool on FreeBSD 4.7.

You can find `anwrap` at `http://packetstormsecurity.nl/cisco/anwrap.pl`.

As a result of cracker tools like `asleap`, THC-LEAPcracker and `anwrap`, Cisco has de-emphasized the use of LEAP, especially for those organizations that can't or won't enforce strong passwords. They now recommend the use of EAP-FAST.

Network Authentication Countermeasures

If you had your heart set on a life of carefree wireless-network use, maybe you're ready to put your head in the oven and turn on the gas. Don't do it. There are some things you can do to protect yourself. Help is on the way.

WPA improves the 8021.1 picture

Because of the WEP problems, the IEEE approved Wi-Fi Protected Access (WPA) as an interim solution to address those problems. WPA is an example of a software or firmware patch and does not require the hardware upgrade that 802.11i does.

The objective of WPA was to bring a standards-based security solution to the marketplace to replace WEP until the availability of the full-blown IEEE 802.11i Robust Security Network (RSN), an amendment to the existing wireless LAN standard.

Chapter 16

Ten Essential Tools for Hacking Wireless Networks

In This Chapter

- Turning on and moving out (with the right laptop computer)
- Hooking up (with a good network card)
- Tuning in (with a high-gain antenna)
- Getting found (via the GPS system)
- Going wireless (with various software tools)
- Looking around (with Google)
- Looking up the rest (with a first-rate wireless reference guide we happen to know)

As with any trade, it's essential to have the right tools when testing your wireless network for security vulnerabilities. Here are ten tools we have found that get the job done.

Laptop Computer

For starters, you've got to a have a good test system — preferably a portable laptop computer. Although it is possible to perform wireless-security testing using a handheld device such as a Pocket PC, the tools available on such devices are limited compared to those on a laptop system.

Due to the multiple operating system requirements of the popular wireless testing tools, we recommend using either a system that can dual boot Windows (preferably 2000 or XP) and Linux (any recent distribution will do) or a Windows-based system running a virtual machine program (such as VMware) on which you can install multiple operating systems. The hardware requirements for systems running a single operating system are pretty minimal given today's standards. A system with a Pentium III or equivalent processor, 256MB RAM, and at least a 30–40GB hard drive should be more than enough. If you'll be running VMware or another virtual machine program, you'll want to at least double this amount of RAM and hard drive space.

Wireless Network Analyzer

To probe deep into the airwaves, a network analyzer is essential. Programs such as Kismet, AiroPeek, and `ethereal` can help you monitor multiple wireless channels, view protocols in use, look for wireless system anomalies — and even capture wireless data right out of thin air.

Port Scanner

A port scanner such as `nmap` or SuperScan is a great tool for scanning the wireless systems you stumble across to find out more about what's running and what's potentially vulnerable.

Vulnerability Assessment Tool

A vulnerability-assessment tool such as Nessus, LANguard Network Security Scanner, or QualysGuard is great for probing your wireless systems further to find out which vulnerabilities actually exist. This information can then be used to poke around further and see what the bad guys can see and even potentially exploit.

Google

It's not only a great reference tool, but the Google search engine can also be used for searching Network Stumbler .NS1 files, digging in to the Web-server software built in to your APs, finding new wireless-security testing tools, researching vulnerabilities, and more. The Google taskbar (downloadable for Internet Explorer, built in to FireFox) makes your searching even easier.

An 802.11 Reference Guide

While performing ongoing ethical hacks against your wireless systems, you'll undoubtedly need a good reference guide on the IEEE 802.11 standards at some time or another. The 802.11 wireless protocol is very complex and will evolve over time. You'll likely need to look up information on channel frequency ranges, what a certain type of packet is used for, or perhaps a default 802.11 setting or two. The Cheat Sheet, the wireless resources found in Appendix A in this book, as well as Peter's book *Wireless Networks For Dummies* are good references that can really help you.

management and communication skills. Your plan is the company's official declaration of what it wants to accomplish and how it wants to do it. Remember: Very few people ever arrive at their destinations without first intending to get there.

At a minimum, your plan should specify the following:

- The roles and responsibilities for everyone involved in the ethical hack.
- The level of involvement of each tester and the importance of her participation in the team.
- The schedule for when the testing will take place. Management may prefer that the testing be done when traffic is low, which might translate into late nights, early mornings, or weekends.

Your security defense budget is likely small, so you need to operate with efficiency and creativity to do more with less. So plan carefully and meet the expectation of the plan. Otherwise, in the future, your management or customer may see security testing as an unnecessary cost.

Not Involving Others in Testing

Often the trick to a successful test lies in observing the details. One such detail is the inclusion of other individuals from your organization in the testing process. Talk to your network professionals. Get people involved up-front during planning. They may help you save time or money by providing insight into the network that you might not have.

Ensure that the testing process is closely monitored by others. Involving others during the process may save you reporting time or may save your hide if you're accused of something you did not do.

Not Using a Methodology

Ethical hacking is different from penetration testing. Ethical hacking is extremely methodical and relies on a method. In Chapter 2, we discuss the concept of the scientific method. You need to adopt or develop a method. Your method should consist of the following steps: planning, testing, and reporting. The method may consist of best practices, such as Open-Source Security Testing Methodolgy Manual (OSSTMM) and Information System Security Assessment Framework (ISSAF). (For more on these terms, refer to Chapter 2.)

deal of effort. Peter has found that writing and getting agreement to a formal report, takes three times as long as the work itself. It really doesn't matter how good your work is if you cannot tell the story well.

Also, report the risk vulnerabilities you've discovered *promptly*. Don't wait until someone exploits the vulnerability or until you report your findings. Your company or customer would have a difficult time proving due diligence if you knew about the vulnerability but did not report it. In these circumstances, not only should you report these items to your boss or customer, but you must also present them with a practical solution.

Before you start writing your report, plan the activities you need to prepare and submit the report. Plan to share your findings with all those with an interest, such as network administrators and your sponsor. You should also plan to share a draft with people. These steps save time.

Your report is one way for you to show the completeness and rigor of your testing methodology. Your peers can review your methods, your findings, your analysis, and your conclusions and decide whether you came to the correct conclusions based on what you report. Some thoughts on reporting:

- Reports should include the following sections:
  - Executive Summary
  - In Scope Statement
  - Out of Scope Statement
  - Objectives
  - Nature of the Testing
  - Analysis
  - Summary of Findings
  - Vulnerability Summary
  - Countermeasure(s) to Control the Vulnerability
  - Conclusion
  - Supporting Documentation
- Reports should include all assumptions regarding the network or system under review.
- Reports should include all unknowns, and they should be clearly marked as unknowns.
- Reports should state clearly all states of security found, not only failed security measures.

• B •

• C •

• N •

• T •

• U •

• V •