Take Control of Your Wi-Fi Security

by Glenn Fleishman and Adam C. Engst

Table of Contents (Version 1.0)

Read Me First 2

Introduction 5

Wi-Fi Security Quick Start 7

Determine Your Security Risk 9

Prevent Access to Your Wireless Network 23

Secure Your Data in Transit 35

Protect Your Systems 56

Secure Small Office Wi-Fi 65

Perform a Security Audit 79

Appendix A: Use WPA Enterprise 92

Appendix B: Password Advice 101

Glossary 104

About This Ebook 111

ISBN 0-9759503-9-8

$10

TidBITS Electronic Publishing

Read Me First

Welcome to *Take Control of Your Wi-Fi Security,* version 1.0.

This ebook is devoted to helping you most effectively secure your home and office wireless network under Mac OS X, Mac OS 9, and Windows XP using common networking hardware. This book was written by Glenn Fleishman and Adam C. Engst, edited by Tonya Engst, and published by TidBITS Electronic Publishing.

You can contact TidBITS Electronic Publishing by sending email to tc-comments@tidbits.com and view the Take Control Web site and catalog at http://www.takecontrolbooks.com/. You can read About This Ebook to learn about the author, the publisher, and the Take Control series. The copyright page contains copyright and legal info.

The price of this ebook is $10. If you want to share it with a friend, please do so as you would with a physical book, meaning that if your friend uses it regularly, your friend should buy a copy. The Help a Friend button on the cover makes it easy for you to give your friend a discount coupon.

We may offer free minor updates to this ebook. Click the Check for Updates button on the cover to access a Web page that informs you of any available or upcoming updates. On that page, you can also sign up to be notified about updates via email.

Onscreen Reading Tips

We carefully designed the Take Control ebooks to be read onscreen, and although most of what you need to know is obvious, note the following for the best possible onscreen reading experience:

- Blue text indicates links. You can click any item in the Table of Contents to jump to that section. Cross-references are also links, as are URLs and email addresses.
- Work with the Bookmarks tab or drawer showing so that you can always jump to any main topic by clicking its bookmark.
- Find more tips at http://www.takecontrolbooks.com/faq.html#reading1.

Printing Tips

Although our layout is aimed at making online reading an enjoyable experience, we've made sure that printing remains a reasonable option. Please review these tips before you print:

- Use the Check for Updates button on the cover to make sure you have the latest version of the ebook and to verify that we don't plan to release a new version shortly. If you want to commit this ebook to paper, it makes sense to print the latest possible version.
- Don't throw out your PDF after you print! You must click the Check for Updates button on the cover to get future updates. The link *must* be accessed from the cover of your PDF.
- For a tighter layout that uses fewer pages, check your printer options for a 2-up feature that prints two pages on one piece of paper. For instance, your Print dialog may have an unlabeled pop-up menu that offers a Layout option; choose Layout, and then choose 2 from the Pages per Sheet pop-up menu. You may also wish to choose Single Hairline from the Border menu.
- When printing on a color inkjet printer, to avoid using a lot of color ink (primarily on the yellow boxes we use for tips and figures), look for an option to print entirely in black-and-white.
- In the unlikely event that Adobe Acrobat or Adobe Reader cannot successfully print this PDF, try Preview; several readers have solved printing problems by using Preview.

Basics

In reading this ebook, you may get stuck if you don't know certain basic facts about wireless networking or if you don't understand Take Control syntax for things like working with menus or finding items in the Finder. Please note the following:

- **Path syntax:** This ebook occasionally uses a *path* to show the location of a file or folder in your file system. Path text is formatted in bold type. For example, Mac OS X stores most utilities, such as Terminal, in the Utilities folder. The path to Terminal is: **`/Applications/Utilities/Terminal`**.

  The slash at the start of the path tells you to start from the root level of the disk. You will also encounter paths that begin with **`~`** (tilde), which is a shortcut for any user's home directory. For example, if a person with the user name **`joe`** wants to install fonts that only he can access, he would install them in his **`~/Library/Fonts`** folder, which is just another way of writing **`/Users/joe/Library/Fonts`**.

- **Menus:** When we describe choosing a command from a menu in the menu bar, we use an abbreviated description. For example, the abbreviated description for the menu command that creates a new 802.1X connection in Internet Connect is "File > New 802.1X Connection."

- **Wi-Fi, AirPort, and wireless networking:** Wi-Fi is an industry term that encompasses three short-range, unlicensed, radio technologies, 802.11b, 802.11g, and 802.11a. Apple calls 802.11b "AirPort" and 802.11g "AirPort Extreme." Regardless of the term, it's all wireless networking.

- **Adapters and gateways:** A standard wireless network has two distinct components: a wireless network adapter (or wireless card) and a wireless gateway (or wireless router). The wireless network adapter is attached to or inserted into a computer and connects to a wireless gateway, which in turn manages the entire wireless network and shares your Internet connection.

Introduction

Just because you're paranoid doesn't mean they're not out to get you.

–Internet security saying

Networking wasn't supposed to be like this. When computer networks were invented, no one anticipated hundreds of millions of naïve users. Nor did they expect crackers, viruses, worms, spam, or spyware. But that's where we've ended up. Most people are clueless about security, and few people devote any time to making their systems secure.

The biggest security risk comes from the simple fact that computers are all networked these days: to each other and to the Internet. Want a totally secure computer? Make sure it isn't connected to the Internet, or to any other computer, and put it in a locked room with an armed guard checking identification on everyone who enters. Not very useful, eh?

Wireless networking, because it makes connecting computers so simple, makes proper security even more critical. Before wireless networking, you could rely on a locked door to restrict access to your Ethernet jacks, and thus to your network. But now, transmissions over wireless networks—because they go through locked doors, along with walls, ceilings, floors, and other obstructions—are easily intercepted by consumer-level equipment just like the gear you use to connect your computers and access point. So anyone within range of your wireless network can connect to it, and, unless you've taken appropriate precautions, wreak all sorts of havoc. And, unfortunately, understanding the reality of wireless security is nowhere near as simple as setting up a wireless network to start.

Our goal in *Take Control of Your Wi-Fi Security* is to bring clarity to the topic; to help you determine how worried you should be about different security problems; and to give you the knowledge you need to lock down your network, protect your data in transit, and secure your systems against attack.

Before we get started, we want to mention a few important caveats:

- We're writing this ebook for individual users with wireless networks at home and for people who run small to medium-sized

office networks, not for veteran network administrators who manage large institutional networks.

- Security, whether you're talking about protecting your car, your home, or your wireless network, is hard, mostly because it's always a battle with another human being. Locking your door with a simple knob lock stops amateur thieves, but keeping more experienced thieves out requires a strong deadbolt. And if you live where burglary is likely, or if you have especially valuable property, you have to think about whether multiple locks, alarm systems, or bars on the windows are also necessary. Unfortunately, the kind of people who break into networks are usually much smarter than garden-variety thieves, and as a result, the security measures you must take to stop them are commensurately more complicated. So, our apologies up front, but some sections of this ebook are inherently quite technical.

- Because every network uses different hardware, software, and configurations, we can't give exact, foolproof, step-by-step instructions for every task we explain. That said, by the time you finish reading this title, you should have the background necessary to configure the networking hardware and software you do have (or are willing to purchase) to the level of security you want to achieve.

We've both been using and writing about various forms of networking for more than 20 years, and we've both set up and maintained numerous wired and wireless networks over that time. And over those years of networking computers together, we've experienced the seedier side of the industry: attacks on our networks via the Internet, password thefts, wireless snoopers, and more. We've shared our experience in many magazine articles and public presentations, and now we look forward to sharing it with you.

NOTE For additional information about wireless networking on the Mac, check out Glenn's *Take Control of Your AirPort Network* ebook (http://www.takecontrolbooks.com/airport.html) and, for more general wireless networking information, the print book we wrote about wireless networking in 2003: *The Wireless Networking Starter Kit, Second Edition* (http://wireless-starter-kit.com/).

Wi-Fi Security Quick Start

You can read the different sections of this title in the order presented here, or you can click a link to jump to a topic immediately. That said, if you're new to the topic of security, we strongly encourage you to read Determine Your Security Risk first to get a sense of how concerned you should be about security.

Determine how worried you should be about security:

- Learn about the three Ls of security: likelihood of attack, liability in the event of loss, and lost opportunity. See Determine Your Security Risk.
- Figure out where you stand on the continuum of people who should be concerned about security. See What You Should Do.

Lock down your wireless network:

- Discover which widely used security mechanisms won't prevent determined attackers. See Ignore These Sops to Security and Avoid WEP encryption.
- Turn on wireless security that is guaranteed to keep intruders out. See Use Wi-Fi Protected Access (WPA) or WPA2 and be sure to read Appendix B: Password Advice.
- Test your network to see how secure it is with Perform a Security Audit.

Protect your data in transit:

- Keep miscreants from discovering your passwords and reading your communications. See Encrypt Email Passwords and Encrypt Specific Files and Messages.
- Armor your Internet sessions inside protected tunnels to keep snoopers from listening to your traffic. See Encrypt Data Streams with SSH, Encrypt Chunks of Data with SSL, and Encrypt All Data with a VPN.

Secure your computers:

- Keep viruses, spyware, and crackers out of your computers. See Protect Your Systems.

Set up secure wireless networking for small offices:

- Make sure your organization's users use good passwords; see Appendix B: Password Advice.
- Lock down your office's network and protect your organization's traveling users with the advice in Secure Small Office Wi-Fi and, potentially, Appendix A: Use WPA Enterprise.

Determine Your Security Risk

Security is something we as a society tolerate, not embrace. Your comfort level with security may vary enormously depending on your background and location. For instance, growing up in rural New York State in the early 1980s, Adam left his car keys in his elderly Dodge Colt when it was parked at home. No one lived within a mile; cars driving by were infrequent, easily seen, and usually announced by the family dog; and a rusty Dodge Colt that needed bits of mouse nest cleaned out of its fuel filter on a regular basis wasn't worth much.

Living in a populous suburb of Seattle a decade later, Adam not only didn't leave his keys in the car (then a shiny, red Honda Civic) when it was parked in the driveway, he also locked the doors. Adam's behavior changed—more paranoid or more realistic, take your pick—because of a different evaluation of the three Ls of security: likelihood, liability, and lost opportunity.

You can get a better idea of where you stand in terms of likelihood, liability, and lost opportunity by answering these questions:

- **Likelihood:** How likely is it that someone will break into your wireless network or *sniff* (monitor) the traffic going across your wireless connection? (See Evalute the Likelihood of Attack, just ahead, for more info.)
- **Liability:** What is the potential liability if someone breaks in to your network, either to monitor your traffic or to use your connection for other purposes, including illegal ones? (Read Determine Your Liability, ahead, for details.)
- **Lost opportunity:** How much money and effort are you willing to expend on the security of your wireless network? (See Calculate Lost Opportunity, further ahead, for details.)

In the rest of this section, we help you answer those questions. We don't want to turn you into a tic-ridden paranoiac. Instead, we want to present a fair discussion of the risks and potential outcomes when you rely on wireless networks.

NOTE Adam wrote an article for *TidBITS* that talks about a theory of privacy and why most people don't give privacy much attention. Read the article at http://db.tidbits.com/getbits.acgi?tbart=05951.

Evaluate the Likelihood of Attack

When thinking about the likelihood of attack, consider two variables: your location and the desirability of the item you're trying to secure. In rural New York, at the end of a dirt road, the likelihood of someone stealing an old, rusty car was low, both because of the remote location and because no one wanted the car anyway. Flash forward 10 years to when Adam had an attractive new car in a busy Seattle neighborhood with lots of strangers driving by, and the likelihood of theft rose significantly.

In terms of wireless networks, Adam lives far enough from the population center in Ithaca, New York, that he and his wife Tonya aren't worried about the potential of a snooper: it would be difficult for someone to access their wireless network without parking in their driveway. In contrast, Glenn and his wife Lynn reside in a moderately dense part of Seattle. One day, soon after younger neighbors started renting the house next door, Glenn flipped open his laptop and spotted his neighbor's wireless network. So, for Adam the likelihood of a security breach is low, whereas for Glenn it's moderately high. If your company is in an office building that holds other companies, the likelihood of someone accessing your network is probably very high.

Consider your location

First consider where you use wireless networks, because location is the primary variable when determining the likelihood that someone would try to connect to your network and snoop. It's likely that you use wireless networks in one or more of the locations in **Table 1**. And when we say "use wireless networks," we're talking about either your own network or networks run by others, because when you access someone else's wireless network, you're still at some level of risk.

Table 1: Likelihood of Snooping in Different Locations		
Location	**Details**	**Likelihood of Snooping**
Rural/far away	In your home and far from other houses	Extremely low
Long-range	Over a long-range, point-to-point link with a wireless ISP or neighbor	Low, due to the directional nature of most point-to-point links
Dense urban or suburban	In your home in a dense urban area or with at least several other houses close by	Moderately high, particularly if you have high-tech neighbors, but actual attacks are unlikely
Mixed-use	In a mixed-use residential and commercial neighborhood	Moderately high, since businesses are more attractive targets and are more likely to use wireless networks
Public-space neighborhood	In a neighborhood near a public park or where people can park on the street	High, since community networks receive constant use by a diverse, anonymous population
Office building	In an office building having multiple businesses or a nearby parking lot within line of sight	Very high, due to proximity and the attractiveness of targets
Roaming	While on the road in airports, cafés, hotels, and other locations	Moderately high, due to the ease of monitoring

Unless you fall into the rural/far away category, or the currently uncommon long-range category, there's a non-trivial likelihood that someone could access your unprotected wireless network or watch your traffic without your knowledge.

Pay special attention to the roaming category. Even if you protect your own network, using your computer while connected to untrusted networks can still put your data at risk. Whether the network is free or for-fee, you have no control over the network-based security precautions, and everyone else using the network may have the ability to see your data in transit on a wired or wireless link.

Determine the desirability of your data

Although location is the most important variable in evaluating the likelihood of attack, you can't ignore the desirability of your data.

If you're a home user who uses the wireless network mostly to connect to the Internet for browsing the Web and sending and receiving email, your data is, and pardon our bluntness, quite dull. It's possible that someone sifting through your network traffic could pull out your credit card or bank account number, if you use online banking, but it's quite unlikely since most such Web sites use SSL to protect the contents of each transaction. In fact, the most likely concern if you're a home user is that your passwords could be captured and used to break into other machines or to send spam through your connection.

On the other hand, if you run a small business and frequently transmit customer credit card numbers between computers on your wireless network (perhaps between databases, or even as part of a backup solution), the desirability of your data is significantly higher than that of a home user's data. A snooper could steal hundreds, if not thousands, of credit card numbers fairly quickly, which is a much better haul than trying to sniff a single home user's credit card number.

While the small business scenario is realistic, think of the desirability (to the right customer) of classified government information. In 2005, when Adam gave a presentation at Los Alamos National Labs (a government research organization involved largely with national security), he learned that wireless networks are entirely forbidden at Los Alamos because the value of their data is too high to risk the possibility of snooping.

Determine Your Liability

Though Adam's income as a teenager was low, the potential liability if the old car had been irretrievably stolen was also low—the car wasn't worth much, and he had alternate transportation from his parents and the school bus. In Seattle, however, not only was the Honda Civic worth a great deal more, but it was also his only form of automotive transportation, and being forced to rely on sketchy public transit for grocery shopping and the like would have been a huge hassle.

When thinking about the liability of having a wireless network cracked, a business might be concerned about whether someone in its parking lot could access its confidential data, such as invoices as they pass between employee computers and a central database, email containing information that might interest competitors, or even customer credit card numbers that might be encrypted over a link

between a company Web server and a customer's browser, but totally unprotected on a local network. The liability of having that data stolen could be the bankruptcy of the company. Home users have fewer worries, of course, but do face a potential financial liability should an attacker steal passwords used for logging in to an online banking account or bill-paying system, for instance.

If you work as a sole proprietor or a company in a number of fields related to bank and credit-card information, health care, and a few other industries, new laws in the U.S. and some existing laws in the European Union and elsewhere may provide for huge financial penalties and jail time for failing to evaluate the liability of your network and take sufficient action. These laws can affect a home worker processing medical claims as a contractor and the world's largest health maintenance organizations alike.

But liability isn't just about the theft of data, and the concerns break down into three categories:

- **Access liability:** What happens if someone uses my wireless network to share my Internet connection?
- **Network traffic liability:** What happens if someone is able to eavesdrop on my wireless network traffic?
- **Computer intrusion liability:** What happens if someone on my wireless network breaks into my computer?

Access liability

Do you want to allow unknown people access to your wireless network? This question doesn't come up with wired networks, since no one installs Ethernet jacks on the outside of a house or office. Since many people believe strongly in the sharing ethic, asking this question isn't unreasonable. Although intentionally allowing people to connect to your network does increase your security risk and the risk that your connection will be used to bad ends, the fact that you're aware of the possible presence of outsiders on your network means that you're probably also more aware of the security considerations their presence engenders.

No matter how you answer that question, a few problems can arise whenever someone accesses your wireless network for friendly, or at least benign, purposes:

- If you have a modem-based Internet connection that you share via a wireless gateway, someone connecting to your wireless network could (and likely would) cause your modem to dial out. That may or may not be a problem in itself, depending on how many phone lines you have, but if you're trying to use the connection at the same time, your performance would suffer badly.
- If you have a cable- or DSL-based Internet connection, performance isn't likely to be a concern most of the time, assuming the unknown visitor isn't uploading or downloading vast quantities of data. However, you may be in unintentional violation of your ISP's terms of service or acceptable use policy by sharing your connection. In the worst-case scenario, your Internet connection could be shut off for that violation.

TIP The only national ISP that we're aware of that condones, allows, and encourages sharing and even reselling any Internet connection is Speakeasy Networks (http://www.speakeasy.net/). They're based in Seattle, and Glenn has used them as a residential and business DSL provider for years. They specifically allow and even encourage—as Glenn confirmed with their CEO during a recent press event—all of their high-speed service to be resold, whether T-1, DSL, or WiMax, a new wireless broadband offering.

Other ISPs may allow resale or sharing of specific kinds of accounts, typically at higher expense. Most allow no sharing of any kind, although few enforce this.

- If you pay for traffic on your Internet connection, whether it's a dial-up or broadband connection, which is quite common outside of the U.S., letting unknown people share your Internet connection could result in a nasty and unexpected bill.
- Although we'd like to assume that anyone accessing an open wireless network would use it responsibly, the possibility for abuse does exist. It's possible that an unknown visitor could use your wireless network to send spam or launch an Internet worm attack, for instance, and while neither would likely hurt you personally all that badly, your ISP might shut you down for being the source of the abuse.

- Lastly, consider the slavering legal hounds of the Recording Industry Association of America (RIAA), those moralistic guardians of a corrupt and abusive industry. If someone were to use your wireless network to share copyrighted songs, it's not inconceivable that the RIAA could come after you as the putative source of the copyright infringement. They've sued universities for their students' behavior and even a deceased 80-something lady who had never owned a computer; how far away are you as a *de facto* ISP, even if your service is free?

RANT! We don't encourage anyone to violate copyright, ludicrous as current copyright law may be thanks to Disney's well-funded lobbying, but we also find the RIAA's tactics utterly offensive. For more on this topic, read J.D. Lasica's well-researched book *Darknet: Hollywood's War Against the Digital Generation* (http://www.darknet.com/).

Your mission, then, is to determine how concerned you are about each of these access-related possible scenarios, after which, of course, you can read the rest of this ebook to address your concerns.

Network traffic liability

You likely believe that most of your private data sits on your computer, that you transmit and receive only limited amounts of sensitive information, and that someone would have to listen at a specific time to capture those bits. The reality of the situation is that we all transmit and receive quite a lot of sensitive data that people with common equipment and widely available software could extract easily from an unprotected network.

Most of the data sent or received over a wired or wireless network is transmitted *in the clear* to anyone able to join or plug into the network. In the clear means that the data is sent in a form that a human being can intercept and then either read directly or convert easily into usable data.

Here's a list of what you might be sending or receiving in the clear:

- Your email account password
- The text of all email messages sent and received
- The contents of any documents sent or received as attachments

- The location and contents of any Web pages viewed
- Your user name and password for any non-secure Web sites (sites that don't use SSL)
- Your FTP user name and password when sent via plain FTP
- Files transmitted via FTP or FTP over SSH (but not SFTP or FTPS)
- The text of most instant messages you send or receive
- The contents of any music or other files you send or receive using LimeWire, Kazaa, or other peer-to-peer file sharing programs
- The IP addresses and port numbers of any connections you make
- The complete contents, including passwords, of telnet sessions
- Timbuktu remote control or file transfer sessions, or VNC remote control sessions (including those via Apple Remote Desktop 2)

These items are *not* sent in the clear:

- The contents of encrypted sessions using SSH, SCP, SSL, or a VPN (described in Secure Your Data in Transit)
- Your POP-based email account's password if your ISP uses APOP
- Timbuktu Pro, VNC (including Apple Remote Desktop 2), or pcAnywhere passwords
- Files and account information transmitted via SFTP (Secure FTP) or FTPS (FTP over SSL), which secure all passwords and data in transit
- Voice conversations and instant messages sent via Skype, which encrypts all traffic
- Instant messages sent using iChat 3 and a Mac OS X 10.4 Tiger Server's Jabber server
- AppleShare passwords (if both client and server have encryption enabled)
- Any secure SSL Web pages (their URLs begin with **`https`**)
- The contents of any email message or file encrypted with PGP or similar public-key encryption technology

WARNING! Even if you close your network through means we describe in Prevent Access to Your Wireless Network to you may still expose data to network crackers and others who can penetrate the basic methods of preventing access. We talk about securing the contents of what you're sending in Secure Your Data in Transit.

Each item that you might transfer in the clear falls into one of three categories: account access information (user names and passwords), information that could be used to track your online steps, and content related to what you say and do:

- **Access information:** Most important is account access information, which, when stolen, presents two types of risk. First, since most people tend to use the same passwords in multiple places, having your email password stolen could compromise a more-sensitive system, like your online banking account. Second, attackers often use a password to one account to break into another account, working their way ever deeper into a computer with the eventual goal of stealing data, causing damage, or using the computer to run an automated program that attacks other computers. In this respect, protecting your passwords isn't something you do just for your own benefit, it's something you do for the benefit of everyone who may be affected if the attacker takes out a server that you use.
- **Online movements:** Similarly, information that tracks online movement doesn't worry either of us, since as journalists, we can always claim we visited a Web site for research purposes. (That might not work if it's a site we visited 1000 times.) But it doesn't take much imagination to see how the fact that a politician had frequented certain sex sites could ruin his career. Again, you probably have a decent idea of whether your online movements could be in any way damaging.
- **Content:** We're pretty transparent people (well, not literally), so there isn't much that we would say or do online that we would worry about someone else reading. We might be embarrassed if the wrong person read the wrong document, but that's it. But what if that document were posted on a widely read mailing list or Web site? Even for us, that could be a problem, and other people might have data that could get them fired, damage their businesses,

humiliate them publicly, or cause lawsuits or divorces. You probably have a pretty good sense of whether or not you're at risk from the things you say or do.

In general, because anything you send or receive could be intercepted and read (text) or used (files and programs), you must accept the notion that everything could be examined or stolen if you're in a location where other people might be able to connect to the network you're using, or even the network your recipient is using.

So what's your liability? Obviously it depends on the data you're transferring, but no one wants their passwords in other people's hands, and we strongly encourage everyone to take some basic precautions that we outline in Secure Your Data in Transit. And if you're more concerned, that section also has solutions for even the most anxious.

Computer intrusion liability

The final form of liability you should consider when thinking about security for your wireless network is what happens if someone uses your wireless network to break into your computer.

NOTE Protecting computers from intrusion via your wireless network isn't fundamentally different from protecting them from intrusion via your wired Internet connection. However, many intrusion programs trust computers on the same local network more than computers on the rest of the Internet, and you must make sure your settings reflect your degree of risk.

We see several types of concerns here.

- **Data theft:** If someone can gain remote access to a computer and its files, she could easily steal sensitive files. All it takes is a few minutes of inattention, or a misconfigured setting, for someone to copy files from your computer. Glenn found this out back in 1994, when his Unix server's password file was stolen (but the passwords weren't cracked, at least) and in 2005 when a Brazilian cracking team almost gained access to one of his Linux servers via Web traffic analysis software—but instead, they just disabled its access to the Internet. And, more recently, Adam was irritated with himself after his ISP asked if he knew that anyone could see files on one of his Macs via AppleShare.

- **Data damage:** You may never know if someone has stolen files from your computer, but you'll certainly realize if he instead vandalized your system and deleted all your files. Worse, some attacks focus on more subtle destruction or manipulation that you wouldn't notice at all. If someone were to tweak Excel spreadsheets with hundreds of numbers in them, could you tell?

> **TIP** A package called Tripwire (http://sourceforge.net/projects/tripwire for open source and http://www.tripwire.com/ for a commercial version) scans your system and creates a cryptographic signature for each file. You then store these signatures on unchangeable media, like a CD-ROM. Each time Tripwire runs, it reports on any changed files, which could help you pinpoint compromises.

- **Exploitation:** Some attacks focus on known bugs in software that allow a remote program or person to infiltrate your computer and take control of some of your software or the entire operating system. Once the attacker has established that level of control, he can install software that acts on his or her command, turning your computer into what's called a *zombie*. Most attacks are aimed at Microsoft Windows or specific Windows software from Microsoft, such as Outlook or Internet Information Server. Over the years, many different bugs have been found that allow attackers to take over a machine; equally as problematic are worms and viruses that may cause damage, replicate themselves, turn the infected computer into a zombie, or all three.

  Zombie attacks against other computers may use a *denial-of-service* (*DoS*) approach, where the attacker tries to overwhelm a computer by sending it huge amounts of data. DoS attacks don't cause damage, per se, but they prevent normal operation and can be difficult to shut down. Zombies are also highly involved in sending spam that sells actual products or that links to fraudulent activity. The latter including *phishing,* in which notes are sent that appear to come from banks or other financial institutions.

  Microsoft keeps patching known holes, but many Windows users don't download and install these security patches, leaving their computers open to further exploitation and infection. One security

problem for both Windows and Mac OS X users is the raw size of the security update downloads: broadband users may have little trouble downloading 80 MB security updates, but downloading one over a dial-up connection could take 6 or 7 hours, and that's if the connection stays up the entire time.

NOTE Glenn once spent a full day watching his network be saturation-bombed with garbage traffic before he could convince an ISP from whose network the attack was launched that he had a serious problem. Glenn finally, with informal advice from the FBI, suggested he might have to sue the ISP and mentioned the FBI; finally, the ISP shut down the offending DSL customer (who was likely the victim of an attack that had turned his computer into a zombie).

The liability for each of these three scenarios—data theft, data damage, and exploitation—is fairly severe; but luckily it's easy to take simple precautions that significantly reduce the likelihood of anything bad happening. Protect Your Systems offers the necessary advice.

Calculate Lost Opportunity

We're stretching a little to find a third L. Lost opportunity, or opportunity cost, is the cost in time, money, and effort in achieving a security goal. The "opportunity"—time, money, and effort—is "lost" because if you spend it on security, you can't spend it on something else.

Obviously, Adam could have reduced the risk with his old car in rural New York even further by locking the doors and keeping the keys in the house, but it wasn't worth doing. The low likelihood and minimal liability made the effort unnecessary. Once he had a new car in an area where car theft was more likely, though, Adam was willing to expend more effort and money: locking the doors when the car was in the driveway, buying and using a brake lock when parking in a Park-and-Ride, and so on. More money and hassle would have provided even greater security—an electronic car alarm, for instance, or a house with a locking garage.

The same situation applies with wireless networking. You can use every available security technique (even the ones that are easily

broken) and in doing so, you'll find that accessing your network is a royal pain. But it will be annoying for would-be attackers as well. Some effort likely makes sense, such as using a password that is relatively easy to type but not easily guessed, and organizations that need even more security can consider a variety of techniques for keeping snoopers out and for protecting data traversing a network that work well, though at a price.

Obviously, the amount you're willing to spend on wireless network security relates directly to your liability. The higher your liability, the more you should be willing to spend, and the more effort you should be willing to expend.

Don't fall into the trap of assuming that a low likelihood of attack means that you can avoid spending time, money, and effort on your wireless network security. It's all about liability, and if the result of an attack could be highly damaging to you, reconsider your willingness to pay for security accordingly.

What You Should Do

Let's combine likelihood, liability, and lost opportunity for a number of sample users to evaluate your real-world risks and determine which sections of this ebook are most important for you to read:

- If you're a home user with no immediate neighbors or nearby public spaces, and if you don't believe your data is particularly sensitive, you don't have much to worry about. At most, read Protect Your Systems to see if you want to take steps to prevent anyone from attacking your computers over the Internet.
- If you're a home user in an urban environment, you should definitely read Prevent Access to Your Wireless Network and Encrypt Email Passwords. If you're concerned about the sensitivity of your data, read the rest of Secure Your Data in Transit as well and consider running through the steps in Perform a Security Audit. It's also worth reading Protect Your Systems just in case.
- If you maintain a wireless network in a business, you should read this entire ebook, thinking hard about your company's risk factors as you go. In particular, in Secure Your Data in Transit, consider

how far you want to go to protect your organization's sensitive data. Also important are Protect Your Systems and Perform a Security Audit because your data is probably more attractive to electronic thieves than the data of a home user. Lastly, be sure to read Secure Small Office Wi-Fi carefully, since it contains information specific to small office needs.

- If you regularly use wireless networks while traveling, be sure to read Secure Your Data in Transit. The more sensitive your data, the more seriously you should consider the approaches in that section.

Prevent Access to Your Wireless Network

Wireless networks weren't originally designed to be very secure. The only encryption available, Wired Equivalent Privacy (WEP), was supposed to work as well as those locks you find on old bathroom doors that can be picked with a paperclip. The designers assumed most people wouldn't have the interest in getting in.

When Wi-Fi became popular, so did cracking techniques and tools, busting WEP's never-strong encryption. Further, most people buying Wi-Fi after the first wave weren't early adopter geek tech-heads. So security options, when available, weren't turned on.

As cracks and flaws evolved, so did replacement technologies, such as Wi-Fi Protected Access (WPA and WPA2), WEP's replacements. You can now reliably secure home and small business networks without much fuss. You just have to know where to start. In this section we look at common mistakes and techniques that don't provide any real security, and then run through how to secure your network with assurance.

Change Network Name and Admin Password

The first step in preventing access to your network is changing some default settings that—when unchanged—make cracking significantly easier. Connect to your wireless gateway with its management software and then:

- Change the network name to something other than the default.
- Change the admin password to a secure password that you'll remember.
- Verify that remote administration from outside the network is off.

Of these three simple tasks, changing the admin password is by far the most important. If you failed to do that, anyone who could connect to your network could also guess the default password (it's usually **`admin`** or **`public`**) and then gain control of your wireless gateway. You wouldn't be locked out forever; a factory reset would blow away any settings changes an attacker made, but you don't want to end up in that situation.

Verifying that remote network administration is turned off is also important. This setting should be disabled by default, but if it has been turned on, anyone on the Internet can attempt to access the management interface for your wireless gateway (and if you've failed to change the admin password as well, you're doubly at risk). There are legitimate reasons to enable remote administration of a wireless gateway over the Internet—for example, you might be in charge of configuring gateways at several remote locations—but unless you have one, keep that option turned off.

Lastly, changing the network name is always worthwhile, partly because it says to any would-be crackers that you know enough to do it (many people don't) and thus your network is more likely to be properly secured. But the main reason to change the network name is because the WPA encryption method uses the network name when generating encryption keys. Because so many people fail to change their network names, keys generated with those default names are significantly weaker and more prone to being cracked than keys generated for networks with unusual names.

NOTE Kudos to Apple in this regard; new AirPort base stations create networks that include the base station's MAC address in the network name by default. Not only does that ensure that every AirPort base station will create a differently named network by default, but also the hexadecimal MAC address looks sufficiently ugly that users are more likely to change it than other defaults, like **`linksys`** (Linksys gateways) or **`tsunami`** (old Cisco gateways).

Now let's look at two techniques you may have heard a lot about—closing your network and restricting access by MAC address—and which companies used to emphasize, but which are almost completely useless against anyone with a few simple tools. After that, we turn our attention to a pair of encryption methods: the older and completely broken WEP, and the newer WPA, which offers real security.

Ignore These Sops to Security

In the real world, people interested in increased security may remove the street numbers from houses or take the company name off the front door. Still others put up large "No Trespassing!" signs. These

approaches don't prevent burglars from breaking in, and they're analogous to several common approaches to securing a wireless network.

We shouldn't be entirely negative, though, since many people may wish to discourage casual access to their networks without worrying about preventing the serious attacker. Closing your network, restricting access by MAC address, and using WEP may be futile when it comes to keeping a determined attacker out of your network, but they will definitely prevent a casual passerby from sharing your Internet connection.

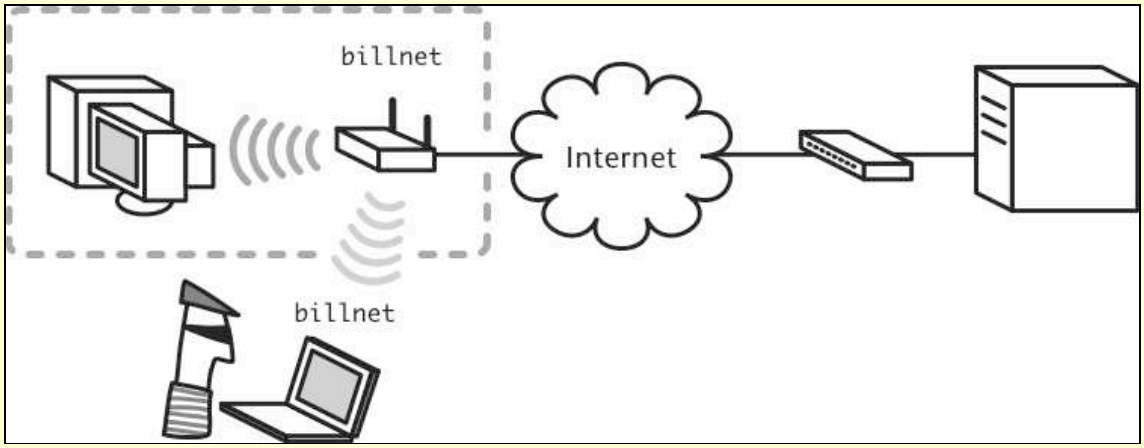
Don't bother closing your network

Most wireless access points enable you to "close" your network, which turns off a message with the network's name that the access point otherwise broadcasts continuously. These broadcasts make it easy for wireless adapters to find and connect to networks.

Some access points call this option a "closed network," and others ask if you want to "disable broadcast name." No matter what the terminology, a closed network's name doesn't appear in the list of available networks in ordinary client software.

Don't be lulled into a false sense of complacency. Although a closed network offers protection from the casual observer, many sniffer programs that monitor wireless networks—from commercial down to open-source freeware—can easily see the name of a closed network whenever a legitimate user connects to it (**Figure 1**). If no one ever connects, it remains hidden, but that's hardly useful.

FIGURE 1



Closing your network prevents casual users from seeing the network name. But with only a little effort, a snooper can determine the network's name.

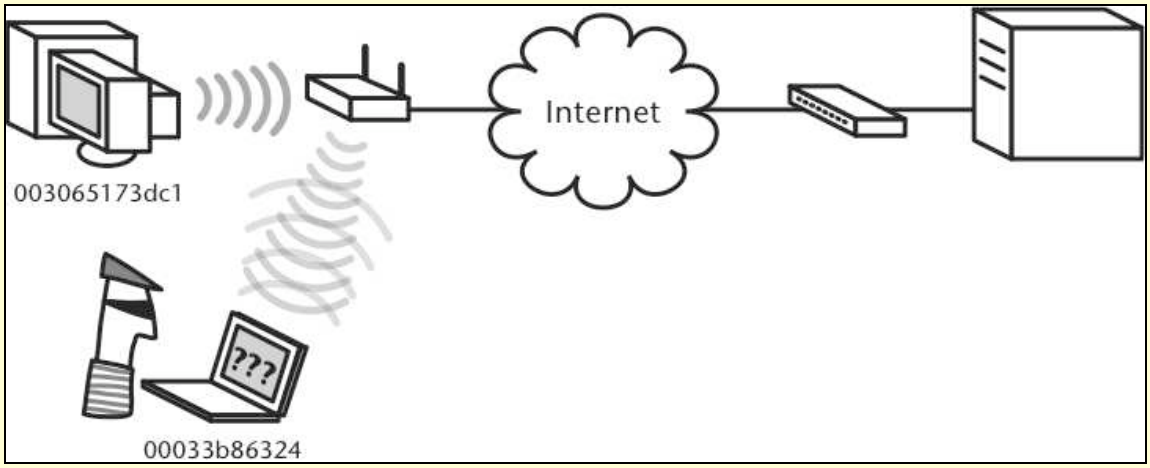
TIP In fact, there is one situation in which closing a network can be useful. If you are mainly concerned about the liability of sharing your Internet connection, and you don't use your wireless network while you're sleeping or at work, closing it prevents even crackers who can see that a network exists from determining its network name and thus connecting to it. Of course, as soon as you come home from work and connect to check your email, the network name will be broadcast in such a way that it can be easily captured, but as long as the network is idle, it will remain secure.

In short, if you don't want average people connecting to your network, there's nothing wrong (other than extra hassle) with closing it, but the only people you're keeping out are those who almost certainly weren't a security risk anyway. That may be worthwhile to you, if your main goal is to prevent passersby from sharing your Internet connection, but you shouldn't consider it real security.

Ignore MAC address-based access controls

Another mostly useless way to restrict access to a network is to allow only specific network adapters to connect (**Figure 2**). Like all Ethernet network adapters, a Wi-Fi adapter is identified by its *MAC (Media Access Control) address,* a unique serial number assigned to every network adapter.

FIGURE 2



Restricting access by MAC address helps keep unauthorized computers out of your wireless network. The attacker's computer isn't allowed to connect to your network because its MAC address isn't authorized.

However, MAC addresses can be *spoofed* (faked to seem like a different address) easily; for more information, see the Wikipedia entry on MAC address at http://en.wikipedia.org/wiki/MAC_address.

This flexibility, combined with the fact that MAC addresses are sent in the clear even on encrypted networks, means cracker can easily see MAC addresses in use and then assign one of those addresses to her equipment. As with a closed network, restricting access by MAC address will keep honest people honest, but it won't do squat against a determined intruder. Worse, if you restrict access to specific MAC addresses, you'll find it annoying to allow a visitor to access your network, since you'll have to enter his laptop's MAC addresses into your Wi-Fi gateway manually and then reboot the gateway.

TIP Many hotspot networks don't use the MAC address to authenticate, but rather use the DHCP-assigned IP address—which is even easier to grab and spoof! Read this disturbing account in *New Architect* magazine at http://www.newarchitectmag.com/documents/s=2445/na0902h/.

Restrict Access and Encrypt Traffic

Enough with those sops! Let's move on to the good stuff. While closed networks and adapter address limitations don't do much good, there is hope. Instead of trying to hide or restrict access, you can use tech-

nology called WEP (Wired Equivalent Privacy) to require that users enter a password to join a network; that same password is used to scramble all the data passing over the network. Without the password, no one can connect to the network or intercept the data.

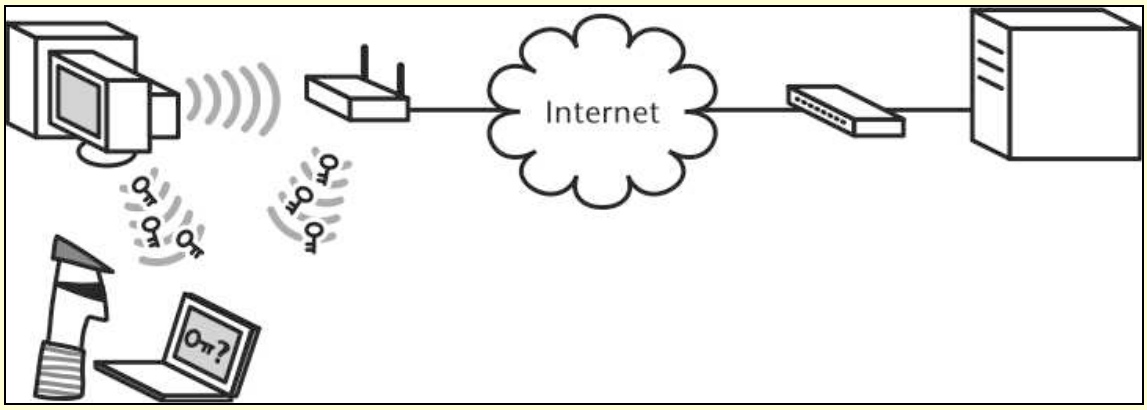
WEP encryption appeared alongside 802.11b back in 1999, but a variety of flaws made it easy to crack. Its replacement is now at hand. We first tell you about WEP, because it's still in wide use, and then about its successors, WPA and WPA2, which we strongly recommend you upgrade to if you currently use WEP.

Avoid WEP encryption

The developers of 802.11b intended WEP to do precisely what the name itself says: offer an equivalent level of privacy to what could be found on a standard wired network. To compromise a wired network, an attacker generally needs to break in to a room and install a network-sniffing program that watches traffic traveling over the wire.

WEP was designed to act merely as a locked door, to keep intruders from penetrating to the wireless network traffic itself; other measures were supposed to bolster this initial line of defense. WEP basically encrypts all the data that flows over a wireless network, preventing attackers from eavesdropping on network traffic (**Figure 3**).

FIGURE 3



Turning on WEP prevents attackers from eavesdropping on network traffic. The attacker can't understand WEP-encrypted traffic without the appropriate WEP key.

Unfortunately, even this relatively minimal protection was crippled because of several brain-dead decisions made on the cryptographic front—some because of the political hot potato that was encryption in the U.S. in the mid-1990s—and because some options were built in

but never enabled by most or all manufacturers. Also, even though WEP still offers some level of protection, most people don't turn WEP on because it's a pain to use.

WEP works by using a "shared secret": an encryption key (up to four per network) shared by everyone on the network. Your wireless network adapter uses the encryption key to encode all traffic before it leaves your computer. Then, when the data arrives, the access point uses the key to decode it into its original form.

NOTE Apple's Mac OS 8.6, 9.x, and X (all versions) allow only a single WEP key to be used on a network instead of up to four, and use something called "Shared" instead of "Open authentication," which renders WEP even more susceptible to cracking. Apple's WPA/WPA2 implementation, discussed later, is just fine.

Users must enter the WEP key manually (and tediously) on every computer that they want to connect to a WEP-protected network. Worse, with most non-Apple hardware, the key is often expressed in the base-16 hexadecimal numbering system in which the letters A through F represent 10 to 15 as a single digit. Most users haven't the slightest idea of how to deal with hex (reasonably enough—that's what computers are for!). If you combine user confusion with the tedium of inventing (on the access point) and entering strings of hexadecimal numbers, you can see why WEP is annoying to use.

You enable WEP in an access point by inventing a sequence of 10 or 26 hexadecimal digits. These correspond to 40 or 104 bits of encryption, which are often called "64" or "128 bits" to include the initialization vector, a 24-bit piece of the encryption key that's not entered by a user.

Some access points have a feature in which you type a passphrase of five or thirteen letters and numbers and then the access point translates that into hexadecimal digits for you. All adapters and access points must use the same length of key on a single network. But this shortcut makes a WEP key even more vulnerable to cracking by reducing the number of possible keys you might have entered.

Aside from the usability problems, how is WEP broken from a security standpoint? Here's a quick rundown of WEP's major flaws:

- **Shared secret:** Every computer on a WEP-protected wireless network needs a set of one to four keys that users must typically type and which can sometimes be read as plain text. The complexity of managing keys makes it easy for an attacker to come by a key through social engineering (asking someone for the key), carelessness (the key written on a piece of paper), or disgruntlement (a fired employee). Most keys are never changed after the first time they're entered. Most network encryption methods, including WPA, can suffer from this problem; it's not unique to WEP.
- **Encryption weaknesses:** Because of how WEP generates unique keys by combining the actual WEP key with a 24-bit number known as an *initialization vector,* the actual key can be extracted relatively easily. The initialization vector isn't used correctly by some manufacturers (who set it to the same number for every packet), or is created in a predictable fashion by others. This results in the reuse of keys, making it easier to break the code, defeating the strengths of the underlying encryption system.

  A more complicated problem is that a subset of all initialization vectors are weak: a cracker can dramatically accelerate the process of cracking a WEP key whenever one of these weak initialization vectors is used. Some adapters and gateways have firmware that prevents the use of weak initialization vectors.
- **Lack of integrity:** While this is usually something that political opponents accuse each other of, integrity in data transmission means that an adapter sending a packet provides some information that allows a receiving station to check that the data that arrived wasn't tampered with in process. WEP's system uses a simple mathematical formula that allows an intruder to rewrite packets and disrupt a network or corrupt data without detection.

These last two problems may sound obscure, but an attacker needs no special knowledge to exploit them; free automated tools perform all the hard work.

TIP For more about wireless encryption problems and solutions, see Glenn's regularly updated security status report at http://wifinetnews.com/weak.defense.html.

Newer cracking software means that even a user with relatively little traffic on his network and/or with little to worry about in terms of interception (passwords but not proprietary data) can't rely on WEP as the only means of protecting the network. The reason? The weak initialization vector mentioned above. Because this problem has been fixed in only some firmware upgrades, it's impossible to know whether your network may suffer from these weak vectors: if it does, a very small amount of data, possibly just thousands of packets, reveals the WEP key in minutes.

With all these flaws, you'd hope that some of the giant brains that develop wireless networking standards would fix the security system. Your hope has become reality: the IEEE engineering group had its 802.11i task group work for years to replace WEP with a forward-thinking, yet backward-compatible, solution that would return WEP to its rightful role as a first line of defense. The first fruits of that work appeared in Wi-Fi Protected Access (WPA), the more complete results later in WPA2.

Use Wi-Fi Protected Access (WPA) or WPA2

Remember those flaws we mentioned in WEP a few paragraphs earlier? With WPA, they're gone. Poof. No more! With 802.11i, phantom security has been replaced with the real thing. The 802.11i standard took years to finalize, but was finally ratified in the middle of 2004.

Before that point, however, the Wi-Fi Alliance had taken matters into its own hands by releasing an interim version of 802.11i called Wi-Fi Protected Access (WPA). The home-user flavor is called WPA Personal; the business version, WPA Enterprise. The Wi-Fi Alliance added the few missing features from 802.11i into a second version of WPA, known logically as WPA2; devices with WPA2 started appearing in early 2005.

WPA2 has a better encryption algorithm available, and it offers some advantages for reducing the delay in hand-off for a mobile Wi-Fi

device, like a Wi-Fi phone that someone is using while walking down a hall between two access points.

When we refer to features found both in WPA and WPA2, we just call it WPA; when we call out a feature found only in WPA2, we use that term.

WPA fixes

Let's take the fixes point-by-point as we did earlier, when we listed what is broken with WEP:

- **Shared secret:** Like WEP, WPA uses a key that everyone who accesses a network agrees on, but rather than using obscure hexadecimal numbers or text characters converted to hexadecimal, the system allows for a plain-text password. In WPA, this is called the *pre-shared key*, and it's described in software as WPA-PSK or WPA Personal. Unlike WEP, the pre-shared key isn't the encryption key itself. Instead, the key is mathematically derived from the pre-shared key. Of course, if someone obtains the pre-shared secret, they can still access your network, but crackers can't extract that password from the network data, as was possible with WEP.

TIP WPA Personal has its own key weakness: short keys based on words found in any dictionary in any language can be discovered through brute force relatively quickly. Making a key at least 20 characters long—pick your favorite obscure song lyric, for instance—overcomes this problem, as does choosing random characters for shorter keys. Changing the network name (SSID) from its default also increases security because WPA Personal uses the network name to create the actual key, and because some crackers compile pre-computed keys based on common access point names and dictionary words.

You can also avoid these weaknesses entirely by using WPA2 Personal with an AES key, as we explain next.

NOTE A more secure way to use WPA and WPA2 that avoids the risk of a shared key is through the enterprise version meant for business use. WPA Enterprise (and WPA2 Enterprise, which works the same) lets each user on a network have a unique encryption key assigned to them when they use a user name and password to login. WPA Enterprise is a form of 802.1X port-based authentication, which we describe in depth in Understand 802.1X.

- **Encryption fixes:** WPA introduces a new kind of key using TKIP (Temporal Key Integrity Protocol), which increases the size of the initialization vector to 48 bits, and ensures that the choice of that number isn't predictable. (All TKIP keys are 128 bits long, too.) This initialization vector change vastly increases the complexity of breaking the encryption system—by several orders of magnitude. Engineers estimate that a key won't repeat for over 100 years on a single device. Even more remarkable, each packet will have its own unique key created by mixing the initialization vector with a master key.

  WPA2 adds to TKIP a much stronger key type that's part of the AES (Advanced Encryption System) set of algorithms. AES keys impose a higher computational burden than WEP and TKIP, but that's not an issue with modern hardware. AES keys aren't vulnerable to the brute force attack on short dictionary word keys mentioned in the tip on the previous page, either.

- **Better integrity:** The integrity of packets is now ensured, eliminating the chance of network disruption. WPA and WPA2 use slightly different methods, with WPA2 employing a stronger scheme.

The only gotcha with WPA is that Wi-Fi adapters that don't support WPA can use only WEP, and packets sent between those devices and a WPA-enabled access point are susceptible to being broken. Worse, only a vanishingly small number of access points allow the use of both WEP and WPA on the same network, which is highly deprecated. Thus, if you have an old laptop whose Wi-Fi card supports only WEP, that forces you to rely on WEP for your entire network, even if every other device on it could use WPA or WPA2. The best option is to upgrade to devices that support WPA.

WPA and WPA2 upgrades

Whether you can upgrade an existing device to WPA or WPA2 depends on its vintage and its original purpose. Let's run through the several types:

- **802.11b adapters sold from 1999 to 2002:** Most of the earliest 802.11b adapters can be upgraded to WPA, but none can handle WPA2: their silicon isn't equipped for it. The problem? Finding the upgrade. Most of the early vendors got out of the Wi-Fi business. Apple still offers WPA upgrades for the original AirPort Card, but the WaveLAN technology on which that card was built was sold first to Lucent, spun off to Agere, sold to Proxim, and liquidated to YDI Wireless. Use Google to see if you can find a firmware upgrade for an old card or, better yet, buy a cheap 802.11g adapter for Mac, Windows, or Linux that has built-in WPA support.
- **802.11b gateways sold from 1999 to 2002:** The original 802.11b gateways fare much worse: the computational power and other factors don't allow for WPA upgrades at all. The original AirPort Base Station and the original Agere residential gateways cannot use anything but WEP. Gateways sold closer to 2002 may be upgradeable to WPA but cannot support WPA2.
- **Adapters and gateways sold from 2003 to present:** Starting in January 2003, however, almost all adapters and gateways—even those that didn't support 802.11g, which was still being finalized at that point—were built to handle WPA and WPA2. Any device introduced since that date either had WPA included or could be upgraded easily to WPA support. Many of these devices' manufacturers released WPA2 updates in 2005, as well.
- **Other Wi-Fi devices:** These devices, which typically attach to Wi-Fi networks to stream music or video, are usually out of date when sold, using only WEP and offering no way to upgrade at all, whether to WPA or WPA2, or even to fix bugs. Check any device you buy to make sure it handles the latest security standards (WPA at a minimum) and that you can upgrade it in some fashion.

Check manufacturers' Web sites regularly, too, as almost all manufacturers provide firmware upgrades every few months to fix minor bugs as well as add these major features.

Secure Your Data in Transit

In the previous section, we explained how to prevent people from accessing your wireless network. However, you may still find yourself in circumstances in which you want protection but restricting access to the network won't help:

- You're sharing your local wireless network or using a shared network on which encryption can't be enabled.
- You're using a public wireless network in a location such as a coffee shop, hotel, airport, or community networking hot spot.
- At least one of your computers can't support WPA, and the use of WEP exposes your whole network to WEP's problems.
- Your employer won't let you use any network except its wired network without encrypting communications to and from your computer.

You have an alternative: you can encrypt the data before it leaves your machine, and have it decrypted only when it arrives at its destination. By creating end-to-end links using strong encryption standards, you can keep your data completely safe from prying network sniffers. Even if people can join your network and reach the Internet—hijacking your link—they still can't see your data. Encrypting your data in transit is a lot more difficult than setting up a closed WEP/WPA-protected network, but it's eminently more sensible.

Tip An added bonus of encrypting data from end to end is that the data you send and receive becomes completely unreadable not just on your wireless network, but also on every Internet link between your computer and the destination machine. That's the reason large organizations typically require their employees to use encryption technology.

We look at five popular categories and methods of securing data in transit, ranging from simple password protection up to full network encryption of all data, summarized in **Table 2**.

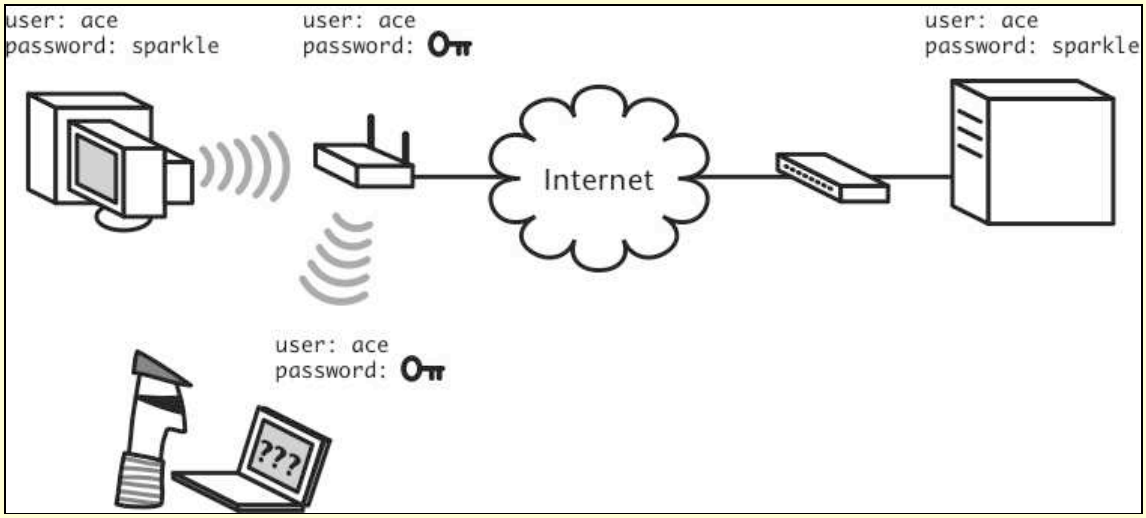
Table 2: Methods of Securing Data in Transit		
Method	**What It's Good For**	**Who Should Use It**
Encrypt Email Passwords	Prevents snoopers from discovering your passwords and using them to access other secured accounts	Everyone
Encrypt Specific Files and Messages	Good for protecting a particular file or email message that's especially sensitive	Anyone with sensitive data to communicate with another person
Encrypt Chunks of Data with SSL	Protecting specific pieces of data, such as credit card numbers, being transferred between programs (like a Web browser and a Web server)	Everyone, when it comes to secure Web sites, and anyone who has easy access to SSL-enabled client and server programs
Encrypt Data Streams with SSH	Protecting data that travels between your computer and your server via a particular protocol	People interested in protecting an entire class of traffic, such as SMTP, POP, or FTP
Encrypt All Data with a VPN	Protecting every last bit of traffic that travels to and from your computer	Business and government users who regularly handle sensitive data, or anyone concerned about transferring confidential data over public wireless networks

Encrypt Email Passwords

Even if you aren't worried that people might read your email, you should worry about protecting your account passwords (**Figure 4**).

NOTE Don't forget that many Web sites use user names and passwords merely for identification and thus don't use secure pages when asking for those passwords. Since these passwords are easily stolen, make sure they're different from your email passwords and passwords to sensitive information. See Appendix B: Password Advice.

FIGURE 4



Encrypting email passwords prevents them from being stolen. In this case, the attacker can't read any passwords protected by APOP and SMTP AUTH.

There are two primary methods of encrypting just your password—APOP and CRAM-MD5—both of which require support in your email client and your mail server, though most modern mail programs have these options. Your ISP or network administrator may have already enabled them on the server side, requiring that you just set an option in your email program.

APOP

APOP (*Authenticated POP*) protects your password when you retrieve inbound email from a POP (Post Office Protocol) server. Instead of sending your password in the clear, APOP sends a unique, per-session token that the server uses to confirm that your email program knows the correct password. The token can't be reused or reverse engineered.

APOP does not encrypt email messages or do anything other than protect your password. We recommend it as a sensible minimum-security precaution; most home users won't need the more significant protections described in the rest of this section.

CRAM-MD5

What if you use IMAP instead of POP for retrieving your email? A technology called CRAM-MD5 can encrypt your IMAP password; if your server supports it, then your email client should automatically use it (you must enable it manually in Apple Mail). However, CRAM-

MD5 isn't particularly secure, which means that for most people, the only way to use IMAP securely is to use it with SSL to encrypt all the IMAP traffic, described in Encrypt Chunks of Data with SSL. Talk to your ISP or network administrator to see if you can rely on IMAP over SSL.

SIDEBAR SMTP AUTH

There's another password associated with email that's not related to encrypting traffic or wireless security. *SMTP AUTH* (the AUTH part is actually an SMTP command), or *Authenticated SMTP,* identifies you to your SMTP server when you want to send outgoing messages. Technically, there's no reason to require authentication for sending email, but Authenticated SMTP has become commonplace in this age of spam, because if an SMTP server requires SMTP AUTH, that prevents a spammer from sending spam through that server. SMTP AUTH typically uses the same user name and password that you use for checking mail via POP or IMAP.

Another benefit of SMTP AUTH is that it can, with some networks, enable you to send email from anywhere on the Internet (such as from a wireless-enabled coffee shop in another city), instead of just from specific network locations that your system administrator has defined. Many wireless ISPs, like Boingo Wireless, offer Authenticated SMTP for outbound email on their networks.

But many residential ISPs (like EarthLink) block all outgoing SMTP traffic on port 25 or redirect all transactions on port 25 to their own mail servers. The workaround for this problem is to switch to sending via port 587, the SMTP submission port. The procedure for using port 587 will vary by email server and email client, unfortunately.

Unfortunately, your SMTP AUTH login information may not be encrypted, which could be a security hole. To find out, first verify that your email program can encrypt SMTP AUTH transactions, perhaps via SSL, and then ask your system administrator, ISP, or email provider about the mail server's capabilities.

TIP You can test for SSL manually by using telnet:

1. Telnet to your mail server:
   - In Mac OS X, launch Terminal and enter the following, replacing mail.server.name with the name of your SMTP server: **`telnet mail.server.name 25`**
   - In Windows XP, choose Start > Run, type **`telnet`** in the Run dialog, and click OK to open Microsoft Telnet. In the Telnet window, type the following (replace mail.server.name with the name of your SMTP server): **`open mail.server.name 25`**
2. Press Return.

   The mail server will respond with a greeting.
3. Now type: **`EHLO nobody.com`** and press Return.

   The mail server will respond with a brief list of methods it supports for starting a session. If **`STARTTLS`** is one of them, the server handles SSL.
4. Type **`QUIT`** and press Return to end the mail session.

This information doesn't guarantee you can make a good SSL connection, but it's a good start.

Encrypt Specific Files and Messages

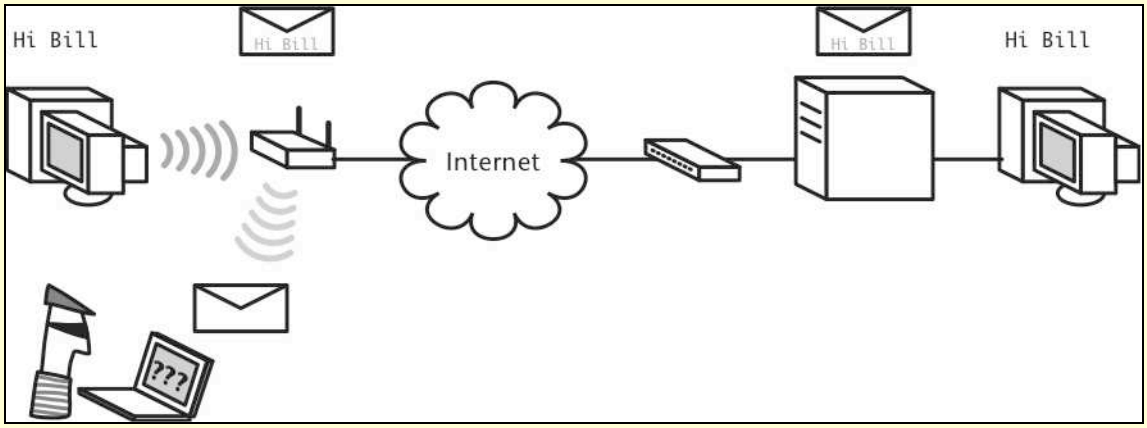
Although we recommend protecting your passwords, there is a middle ground between encrypting passwords and encrypting all your data—using content encryption on specific files and email messages. This approach lets you protect the pieces of content that you feel are the most sensitive.

Content encryption makes it almost impossible that anyone other than your intended recipient could read the file or email message, even if they obtained access to your machine or a mail server between you and your recipient. That's because when you encrypt content manually on your end, it usually requires the recipient to decrypt it with a manual action on the other end.

NOTE If a bad guy obtains access to your recipient's machine, however, all bets are off with regard to whether or not your content can be read. Your recipient may have decrypted it and stored the plain text version already, or she may have weak passwords that enable the attacker to use information on the machine (the name of the hard disk, for instance) to guess the necessary password and decrypt your message.

The most popular software that encrypts the contents of messages or entire files is PGP (Pretty Good Privacy). PGP uses public-key cryptography to secure a message so only the intended recipient can read it (**Figure 5**).

FIGURE 5



Encrypting email messages and specific files with PGP prevents crackers from reading just those messages or files. In this case, although the cracker can see an encrypted copy of the message, he can't read the contents.

NOTE The PGP software was sold by its developers to Network Associates. In 2002, Network Associates sold it back to PGP Corporation (http://www.pgp.com/), which has released version 8 (essentially a maintenance release) and version 9 (a more comprehensive overhaul). PGP 9 also works with POP, IMAP, and SMTP to secure connections using SSL without reconfiguring a mail application!

TIP An open-source alternative to PGP is GPG or GNU Privacy Guard (http://www.gnupg.org/); a Mac version is in development (http://macgpg.sourceforge.net/). GPG works with most files created by newer versions of PGP and vice versa, but read the GPG FAQ if you plan to use the two together.

Understand public-key cryptography

In public-key cryptography systems, each user generates a pair of keys, one public, and one private. Using combinations of those keys, users can sign files or messages to prove that they sent them and can encrypt files or messages so only the intended recipient can open them. The keys work together like puzzle pieces—if someone encrypts something with your public key, only your private key can open it. And, if you sign something with your private key, only your public key can verify that you signed it.

Signing and verifying, encrypting and decrypting basics

As an example, assume that Glenn and Adam set up PGP so they can exchange encrypted drafts of this title without concern about industrial spies from other publishers sneaking a look at the drafts. (In reality, we're nowhere near that paranoid.)

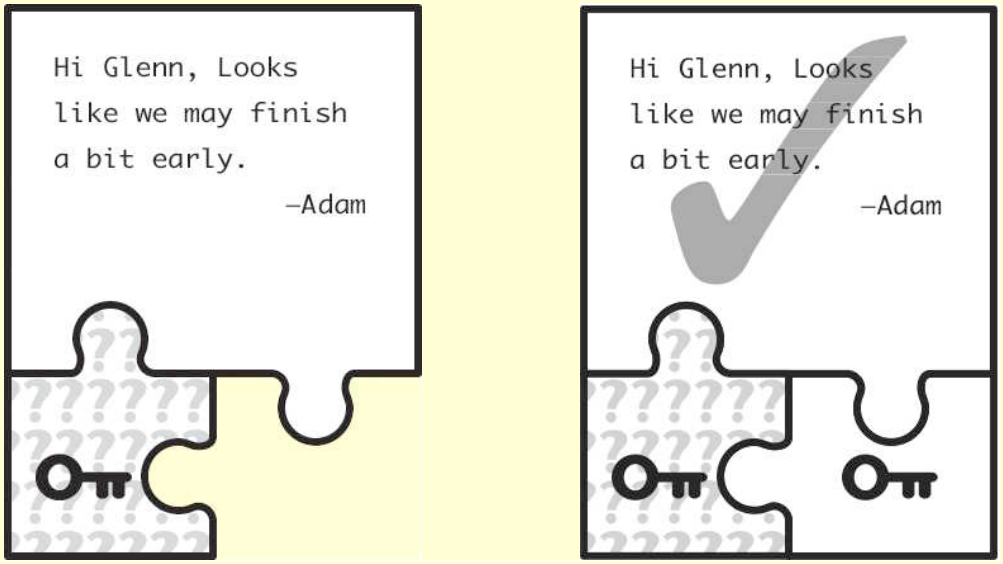
The first step in public-key cryptography is to generate a public key and a private key. Along with the keys, you must generate a pass-phrase that enables you to decrypt and unlock your own private key when you want to use it. Adam and Glenn both run through these steps, so they each have public and private keys, and then they share their public keys with each other (see Distributing keys). Now, here's how they can use their keys.

Adam wants to send Glenn an extremely important email message regarding the book schedule. Adam's not worried about someone else seeing this message, but he does want to make certain that Glenn believes it comes from him, and hasn't been forged by some joker on the Internet. (Email forgery is remarkably simple, though doing it in such a way that it can't be traced easily or identified as a forgery is more difficult.) So Adam signs the message using his private key.

When Glenn receives the message, he can read it with no extra effort, but to verify that it did indeed come from Adam, he uses Adam's public key to check the signature. When they match, Glenn knows

that the message is legitimate (**Figure 6**). Had someone used any other private key to sign the message, it wouldn't have matched with Adam's public key and the signature verification would have failed.

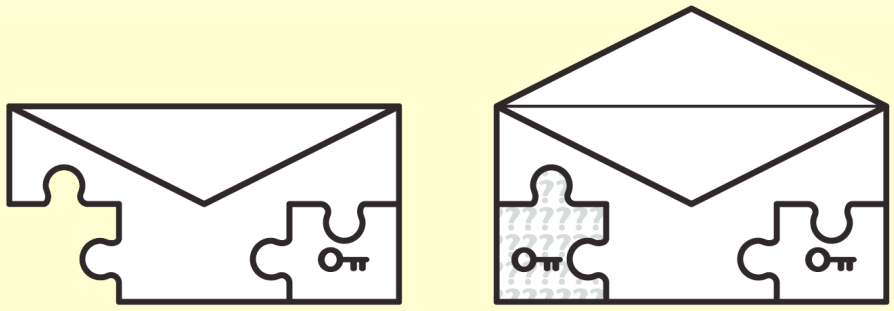
FIGURE 6



Adam signs the message with his private key (left), and Glenn uses Adam's public key to verify the message (right).

Next, assume that Glenn wants to send Adam a draft of the book, but because he's worried that one of his neighbors may be eavesdropping on his wireless network traffic, he decides to encrypt the file before sending it. This time, Glenn uses Adam's public key to encrypt the file, and then sends the file along. When Adam receives the file, he uses his private key to decrypt it (**Figure 7**). If someone were to intercept the file and try to decrypt it, she couldn't because only Adam's private key can decrypt files encrypted with his public key.

FIGURE 7



Glenn encrypts the file with Adam's public key (left), and Adam decrypts the file with his private key (right).

TIP You can see from this example how important it is that you keep your private key safe and don't share it with anyone. If someone were to learn your private key, that person could forge your digital signature and could decrypt any encrypted information sent to you. In reality, if your private key were compromised, you'd have to revoke its public key partner to forestall anyone from using the public key.

Distributing keys

The fact that public keys can be shared without jeopardizing encryption is what makes the public-key cryptography method unique. But sharing public keys is also an Achilles heel: how do you distribute your public key and receive keys from others for the first transaction? You can send it to people in email; include it in your email signature; put it on your Web site; or post it to a public directory, called a *keyserver* (such as keyserver.pgp.com, which is available from within PGP).

Although these methods all work, none are watertight, because someone bent on impersonating you could forge mail from you or post a key to a keyserver while pretending to be you. Once the fake key is out in the wild, revoking it is tricky.

The solution is for people to exchange keys in ways that ensure the other party's identity. For instance, Glenn and Adam could have created their public keys while at lunch together; being able to see the other person is as much verification of identity as is usually required. Slightly less sure, but more reasonable, is using a telephone or fax machine; in those cases you don't read out or write the entire public key (which is way too long for accurate transcribing). Instead, you convey a shorter sequence of letters and numbers that verifies to the other person that the public key you've sent them is indeed yours—the sequence of letters and numbers is called a *fingerprint*. Some people put their fingerprints in their email signatures, assuming that a recipient can email them to verify identity.

TIP PGP offers a neat fingerprinting method: it associates unique words with each number from 0 to 255 (listed in hexadecimal as 00 to FF) to make it easier to read out. Glenn's fingerprint starts "soybean drunken stormy uncut Oakland." Sounds like Beat poetry.

Luckily, although it can be tricky to verify that a public key does indeed belong to a specific person, the worst possible outcome is that someone could distribute a new public key under your name, thus bringing documents ostensibly signed by you into question. But when somebody sends you an encrypted file or message that uses this fake public key, you can't decrypt it with your private key. This should alert you to potential problems, but your security is intact.

Once you have a public key for someone and have verified who she is, you can exchange messages for the life of the key. Many public keys are set to expire on a certain date for additional security.

Use PGP

Special software is necessary to sign or encrypt (and verify or decrypt) files and messages using PGP. PGP Corp.'s Desktop Home 9 is a 30-day trial: after 30 days, either you need to purchase a license for commercial use (such as in a small personal business or corporation), or you can continue to use just its encryption/decryption features for personal and non-commercial purposes (http://www.pgp.com/).

The commercial version comes with various levels of technical support and includes mountable, encrypted disk images, email proxies for secure sessions and rules-based encryption, and secure instant messaging. Using the email proxy lets you encrypt messages based on, for instance, putting "[confidential]" in the subject line.

Encrypt Chunks of Data with SSL

SSL (*Secure Sockets Layer*) was initially developed to secure financial transactions on the Web, but it is now widely used to secure Internet transactions of all kinds. Every time you see an **`https`** URL, you're using SSL in your Web browser to secure the communication between your browser and the remote Web server.

SSL solves the "shared secret" problem found in WEP by using an expanded version of public-key cryptography (see Encrypt Specific Files and Messages, earlier). Instead of requiring that people agree on a secret (the encryption key in WEP) in advance or requiring that public keys be published, an SSL-equipped browser and server use a trusted third party known as a *certificate authority* to agree on each other's identity. (SSH depends on fingerprinting for the same effect, although you can also store public keys on computers that want to create secured tunnels.)

Typically, SSL is used for short session-based interactions, like sending credit card information via a Web form. But it can be used to encrypt email sessions (sending and receiving) including the entire contents of email from the client to the server, for FTP (in a form known as FTP over SSL or FTPS), and for many other transaction types. SSL works for any Internet service that uses TCP for data exchange in which every packet is designed to be received or retransmitted if it doesn't reach its destination. (For example, you can use SSL for instant messaging, but not for RealAudio music playback, although there are some workarounds.)

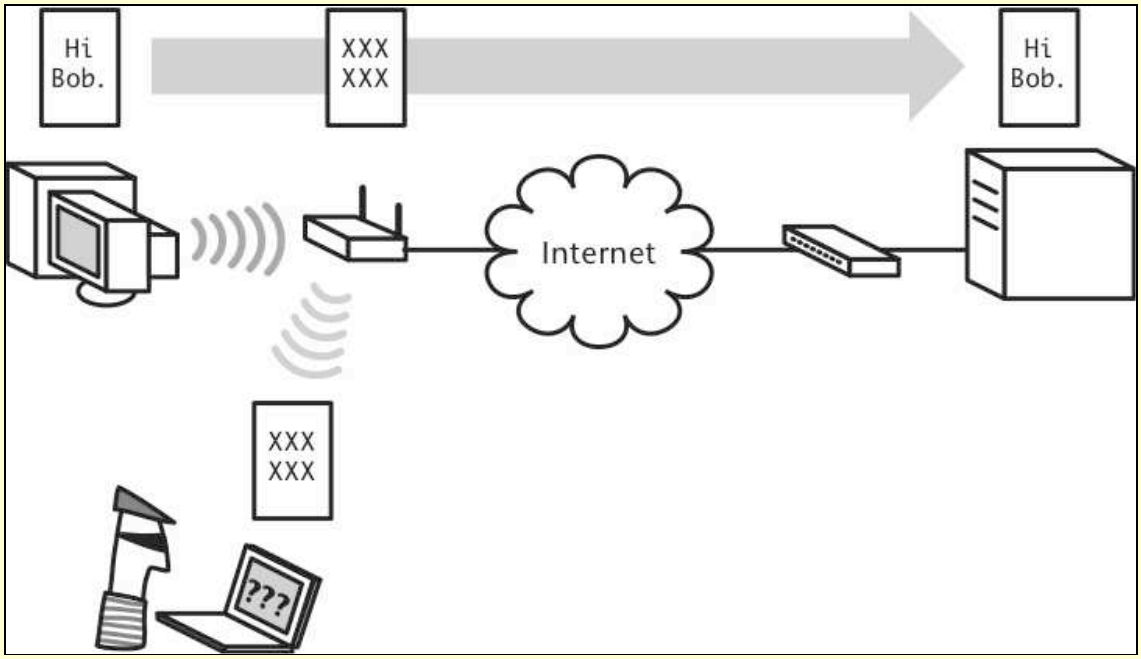
Because a third party can verify SSL, you don't have to rely on trust (blind or confirmed) as you do with SSH.

You can also work with SSL where there's no certificate authority, just a *self-signed certificate*. This certificate isn't signed by another party, but for certain cases, like a private mail server, it's good enough. Some programs don't like these self-signed certificates, and might cause you grief.

NOTE SSL is now widely used because the underlying patents expired. Often, the developers replace the name SSL (Netscape's coinage) with TLS or Transport Layer Security. TLS is newer and has minor technical differences from SSL, but in common parlance—as opposed to among software developers—the terms are used interchangeably. We tend to use the term SSL simply because it's better known. We expect TLS will replace SSL as a standard term over time.

Unlike SSH, in which you can connect any two arbitrary ports through a tunnel (port to port), SSL works from program to program, with the client encrypting data and the server decrypting it (**Figure 8**). But, just like SSH, all data sent over that tunnel, including passwords and all sent and received content, is securely encrypted.

FIGURE 8



Encrypting select transactions with SSL protects just the contents of that transaction. In this case, the attacker can't read any of the traffic encrypted with SSL, but potentially could read other traffic.

Understand how SSL works

When you connect to an SSL-protected Web page, your Web browser and the remote Web server must negotiate the exchange of keys.

Your browser has a preinstalled list of digital signatures from *certificate authorities,* which are companies that confirm the validity of a given certificate being attached to a given server by IP address. When your browser connects to an SSL server, it receives the server's certificate, and uses its built-in list to confirm its validity. Browsers warn you if there's no third-party verification or there's a mismatch.

If the certificate is valid, the browser uses information in it to encrypt a session key, which can be used with no worries that it was intercepted because of the third-party check. (After the certificate is exchanged, the process is very much like PGP for creating the session.)

With an email client and server, a similar transaction happens. The email client requests a secure connection; the mail server responds with certificate information; and the two exchange keys before exchanging mail.

Use SSL where possible

In contrast with PGP and SSH, when it comes to SSL, you can typically encrypt the data sent and received by any SSL-equipped application without entering passwords or any other convolutions. The client software handles communication just as it would with an unencrypted connection. While SSL encrypts the connection without a password, this approach doesn't bypass authentication for the Web site or other service: you must still provide a user name and password to access your bank account, for instance, but that user name and password are secured from outside view.

With the Web, modern browsers and servers use SSL when necessary (assuming the webmasters have set it up properly). You can tell when you are viewing an SSL-protected Web page because there's often a little closed-lock icon in one of the extreme corners of the window. Also, look for the telltale sign in a URL: instead of the URL starting with **`http`**, it starts with **`https`**.

Many email programs support SSL, and turning on SSL simply requires that you set an option, often hidden in an advanced configuration dialog. Unfortunately, not all mail servers support SSL, and of those that do, not all handle SSL in the same way. As a result, some SSL-capable mail servers aren't compatible with some SSL-capable email programs. To learn if SSL is an option for protecting your email, check with your ISP or system administrator, or install a mail server that is compatible with the software you've chosen to use.

TIP PGP Desktop 9 and Desktop Home 9 include an email proxy service that puts the burden for creating an appropriately configured SSL connection on the PGP software. This seems to work more reliably and with less frustration than working out the details ourselves. Even better, the PGP software can monitor for email connections, intercept them, and automatically configure itself—zero setup.

You can secure FTP with SSL in free or commercial software, as long as the FTP server handles SSL connections. On the free side, check out Glub Tech (http://www.glub.com/) to find a Java-based, secure FTP client. On the commercial side, see the article "Secure FTP 101"

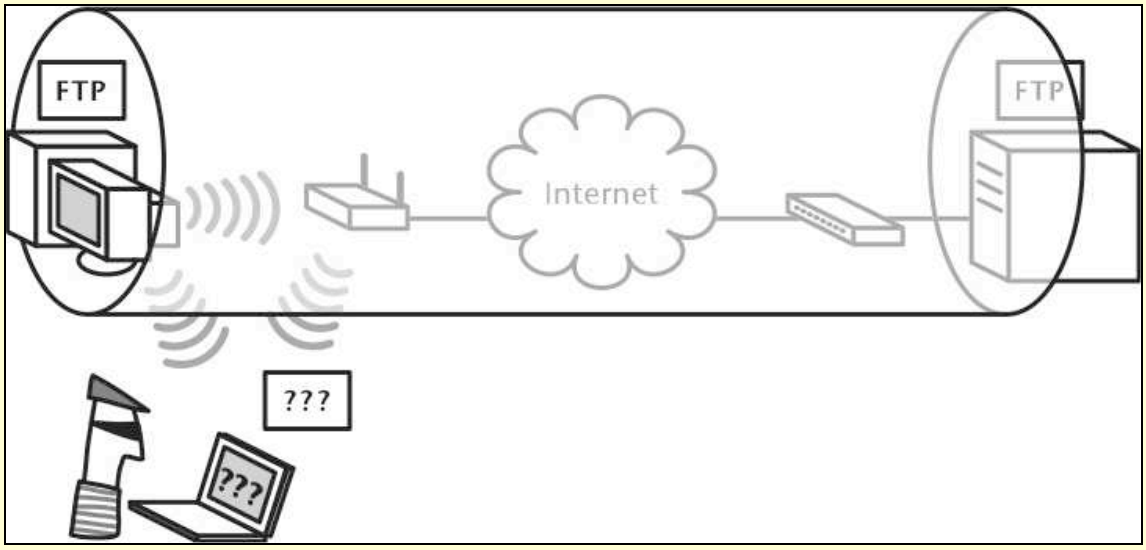
for an overview of secure FTP and a list of commercial software (http://www.intranetjournal.com/articles/200208/se_08_14_02a.html). Also, various software companies have added SSL support to their FTP client software.

TIP A Unix and Windows package called Stunnel lets a system administrator add SSL to practically any service by wrapping SSL around existing server software instead of requiring a different FTP or email server (http://www.stunnel.org/).

Encrypt Data Streams with SSH

SSH (*Secure Shell*) was originally created as a way to establish encrypted terminal sessions that *tunnel*—or create simple end-to-end connections—between a client computer and a server computer (**Figure 9**). SSH was necessary because the telnet protocol sent all information in the clear as plain text, allowing any network snooper to grab data.

FIGURE 9



Creating a secure tunnel with SSH protects the data inside the tunnel. In this case, the attacker can't read any of the FTP traffic in the SSH tunnel.

SSH has expanded far beyond this original purpose. It now lets you create tunnels for any kind of TCP (though not UDP, a lower-level Internet protocol) protocol, whether POP, SMTP, Web, or even Timbuktu Pro. It does this with a trick called *port forwarding*, which

connects a local port on your computer with a remote port on a server.

With SSH, you're protecting both passwords and all the content that you send or receive via the Internet services you choose to tunnel.

TIP After years of working with SSH, we've decided that it's most appropriately used for SFTP (a special form of FTP that combines SSH with FTP for both data and the control connection), terminal sessions, and Timbuktu Pro. For email and other kinds of traffic, SSL is now simpler and superior in most respects. See Encrypt Chunks of Data with SSL.

Understand how SSH works

SSH encrypts the entire contents of any session, and it's considered highly secure. SSH doesn't, by default, use outside trust: the initial exchange between a server and a client to set up a trusted relationship for future sessions requires either blind faith or the use of a confirmation code, also called a fingerprint.

NOTE An SSH server generates a fingerprint for its encryption key, and when you connect for the first time from a client, you can double-check that the fingerprint your client sees is identical to the one on the server. If you run your own server, you can retrieve the fingerprint yourself (see the documentation for OpenSSH). Otherwise, ask your network administrator.

NOTE SSH uses public key encryption, as described earlier, as the first step of a session. After trust is established by exchanging public keys, a session is started by encrypting a much shorter symmetric session key with a public key. The server and client can safely confirm the shorter key. A shorter key speeds encryption for real-time data transfer, and symmetric encryption is much faster than public-key encryption, so public-key encryption is generally used only to establish a random symmetric session key. This approach provides the advantages of public key cryptography without requiring all the computation of a pure-public-key implementation.

Port forwarding with SSH involves connecting a TCP port on your local computer with a port on a remote machine using an encrypted SSH tunnel as the connector. For instance, if you want to retrieve email via an SSH tunnel, you first set up the tunnel between the POP (Post Office Protocol) port (110) on your computer and the remote server's POP port. Then you configure your email program to retrieve email from IP address 127.0.0.1, which is a generic alias for your local machine, on that same port 110. SSH can establish multiple tunnels, such as POP and SMTP, with a single command.

TIP Using low ports (under 1024), such as 110 or 25, requires an administrator account under Mac OS X, and root-level access on other Unix and Linux systems. Because you'd probably connect software servers on your machine only if you were an administrator, this requirement isn't a real problem. You can avoid needing this level of permission if you use ports numbered 1024 or higher.

The SSH software intercepts requests for connections on that port from your mail program and forwards those connections, securely encrypted, to the mail server you specify; responses pass along the same encrypted tunnel.

Fortunately, newer applications that secure their connections with SSH avoid the complexity of setting this up. For arbitrary programs, you still need to know this level of detail, but a modern FTP client might require just selecting SFTP instead of plain FTP.

SIDEBAR FTP OVER SSH, SFTP, AND FTPS, OH MY!

There are three popular kinds of secured FTP, two of which use SSH and one uses SSL. This leads to confusion because of how the acronyms were formed and what's actually secured.

- **FTP over SSH** secures only the control channel, or the part of the FTP transaction that involves sending the user name, password, and commands. The data portion is sent in the clear. FTP over SSH is the only one of the three standards that can work with an arbitrary client and server that aren't running special FTP software, because it encrypts just the single control channel. The other two methods require coordination.
- **SFTP** (Secure FTP) uses the sftp program and sftp-server software to communicate using SSH, encrypting both control and data connections. This preferred approach is widely available. Mac OS X's built-in Remote Login option in the Services pane of the Sharing preference pane enables the sftp-server software. SFTP can work with any arbitrary server that you agree to connect to.
- **FTPS** (FTP over SSL) uses SSL to secure a connection. This method is used much less frequently at the moment because of a lack of client support. FTPS requires the client FTP software to accept and approve a certificate from the server, which can add complexity.

To add confusion, there's also secure copy (scp), a Unix program generally available only on the command line that uses SSH to perform secure file transfers.

The main drawback of SSH is that you must have access to a server that can run SSH or SFTP on its end of the connection—usually true for Unix systems (including Mac OS X) but not true of Microsoft Windows. (It takes two to tango in the SSH tunnel.) You may be able to avoid this problem by running your own SSH-equipped or -capable servers or by working with an ISP or a network administrator willing to set up the connection.

Another issue is that SSH works well only for services that need single ports and use TCP. When you get into more complex arrangements or UDP packets—often used for streaming media—you need to pursue software or services with encryption built in, or something more comprehensive like a VPN (see Encrypt All Data with a VPN).

TIP If SSH isn't an option on your servers, you can try services such as Secure-Tunnel.com (http://www.secure-tunnel.com/) that offer for-fee SSH tunneling services as well as what the company calls "private surfing" using SSL encryption described later in this section.

Find software with SSH built in

Setting up an SSH connection requires a program that can talk to a remote SSH server.

SSH client and server software:

- A great general list of free SSH software is available at the aptly named FreeSSH.org site.
  http://freessh.org/
- For Windows-specific freeware and shareware for making terminal connections and responding to SSH requests from clients, check out the OpenSSH Windows list.
  http://www.openssh.com/windows.html
- SSH Communications Security offers a non-commercial version of its SSH software for Windows. It also offers commercial software.
  http://www.ssh.com/support/downloads/secureshellwks/non-commercial.html
- You can also obtain commercial SSH packages from F-Secure and VanDyke Software.
  http://www.f-secure.com/
  http://www.vandyke.com/
- Mac OS X's Terminal allows command-line SSH sessions. To enable an SSH server in Mac OS X, open the Sharing pane in System Preferences, click the Services tab, and select Remote Login.

SFTP client software:

- Mac OS X users can use SFTP with Interarchy, Transmit, or Fetch.
  http://www.interarchy.com/
  http://www.panic.com/transmit/
  http://fetchsoftworks.com/
- Windows users can use an FTP program from the open-source world called WinSCP.
  http://winscp.sourceforge.net/eng/

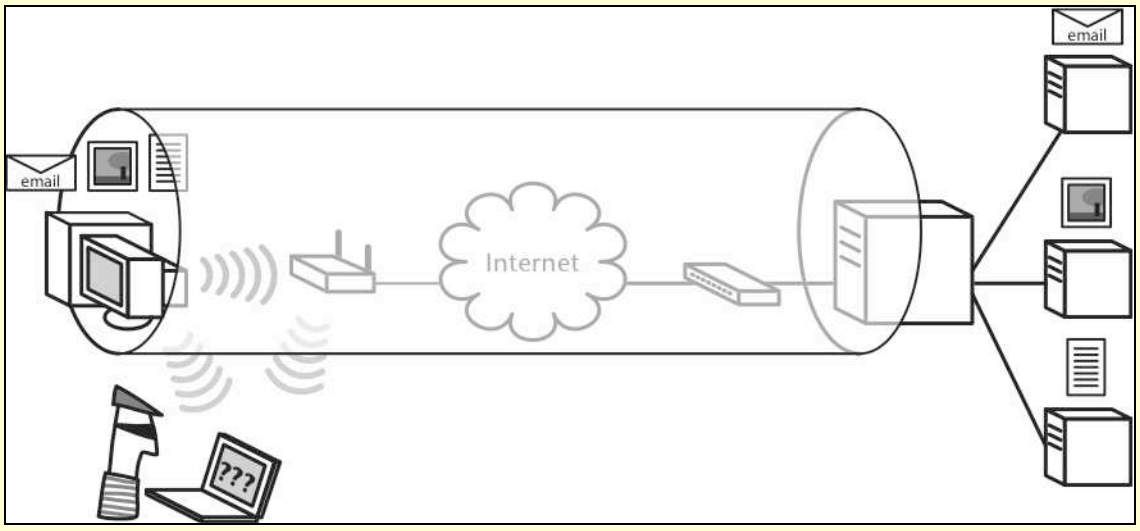
Special programs:

- Programs such as Timbuktu Pro 8 for Mac OS X can use SSH to communicate with other copies of themselves running on other computers.
  http://www.netopia.com/software/products/tb2/

Encrypt All Data with a VPN

As you've undoubtedly noticed, all the encryption solutions we've discussed so far are specific to a type of Internet service, specific files or messages, or certain software. Why not just encrypt everything? For that you need a *VPN,* or *virtual private network*. VPNs are the ultimate solution for securing data because they create an encrypted pipe, called a *tunnel,* that carries *all* the traffic between your computer and a VPN server—it's essentially a secure extension of a network. Since the data you send or receive—email, FTP, Web, and anything else—between your computer and the VPN server is encrypted, you don't have to worry about an intruder breaking into your wireless network. Even if someone were to break in, she couldn't decrypt the tunnel that carries all your communications (**Figure 10**).

FIGURE 10

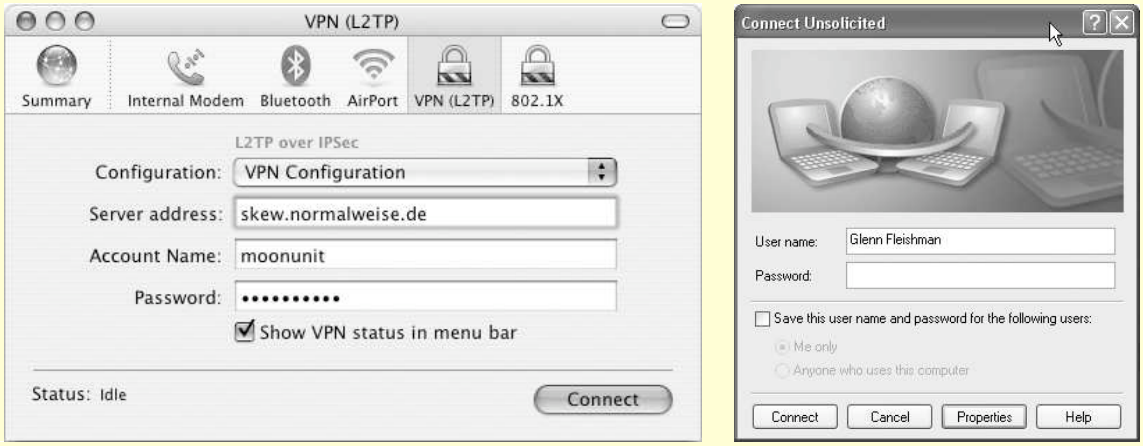


Encrypting all your traffic in a VPN tunnel offers the most complete level of protection. Here, the attacker can't read any traffic at all because of the encrypted VPN tunnel.

Three popular protocols are used for VPNs:

- **PPTP (Point-to-Point Tunneling Protocol):** Microsoft developed PPTP, so PPTP client software ships with most versions of Windows. In addition, PPTP is built into Mac OS X 10.2 and later, and PPTP clients are available for Unix and Linux. SSL-based VPNs are also available.
- **IPsec (short for "IP security"):** Security experts consider IPsec more robust than PPTP and about the same as some flavors of SSL VPNs, and IPsec client software is widely available for Windows, as well as built into Mac OS X 10.2 at the command line and into 10.3/10.4 in the Internet Connect software (**Figure 11**).

FIGURE 11



Connecting to an IPsec VPN in Mac OS X (left) and Windows XP (right).

NOTE IPsec used with VPNs is technically called *IPsec-over-L2TP,* because *IPsec* is an encryption protocol, while *Layer 2 Tunneling Protocol* is the method of running IPsec over an Internet connection. We've also seen it called *L2TP-over-IPsec* for reasons we can't explain.

VPNs using IPsec used to have difficulty on wireless networks that created their own private addresses using NAT (Network Address Translation). However, most access points have been upgraded to pass IPsec VPN traffic through correctly. If your access point doesn't support IPsec, see if the manufacturer offers a firmware upgrade, or buy an access point that can handle it.

- **SSL:** SSL-based VPNs require special client software, but the OpenVPN open-source project has free and well-made graphical

interfaces for Mac OS X and Windows, and many flavors of Unix and Linux, among other platforms (http://openvpn.net/).

A VPN requires both a client and a server, and servers used to be phenomenally expensive. Fortunately, several affordable options are now available, from Internet-based VPN services to low-cost VPN servers that are part of wireless gateways. We discuss these options in Secure Small Office Wi-Fi.

We recommend using a VPN if it's practical and affordable because of the combination of simplicity, complete protection, and peace of mind that it offers.

Protect Your Systems

One part of security is protecting your data in transit; the other part is protecting your systems—your computers, any Internet servers you run, your wireless gateway, and so on—from online intruders. Because wireless networks potentially expose your systems to attackers who would never have the same kind of access on a wired network—unless they broke into your house or business—you need to exercise greater care when protecting your computers on wireless networks.

You can secure your computers against snooping or attack in two ways: an active firewall or network address translation. You can use them separately or, for additional security, combined. And of course, it's essential to run current anti-virus software, particularly if you use Windows. But first, why worry?

Get Paranoid

You might think that you don't need to protect your computers, but, unfortunately, there are seemingly hundreds of thousands of bored, amoral people out there, constantly and automatically scanning large blocks of Internet addresses for weaknesses. These days, it can be only a matter of minutes after a computer first receives a public IP address before the first attack is launched against it. Most of these attacks are entirely automated using scripts deployed by *script kiddies*, or inexperienced crackers who use prefabricated software.

These attacks focus on known bugs in software that allow a remote program or person to infiltrate your computer and take control of some of your software or the entire operating system. Once the attacker has established that level of control, he can either destroy your system or install software that attacks other computers, turning your computer into what's called a *zombie*.

Don't assume that attacks necessarily come from people. It's far more likely that a worm that's already taken over someone else's machine will attack your computer. Worms propagate viruses that in turn propagate worms. The virus may also cause other damage or turn the computer into a zombie for later attacks.

NOTE We're not kidding about being infected within minutes of turning on a new computer. The SANS Institute runs an ongoing survey of how long it takes an unpatched Windows XP computer to be infected; the latest data shows an improvement in the average time to infection of 20–30 minutes. The minimum time was under 20 minutes as recently as a year ago. For the latest data, see http://isc.sans.org/survivalhistory.php. This isn't just theory: Glenn was testing a Windows XP Home laptop in August 2003: he powered the machine up and had just run the browser to download patches when the machine was infected and rebooted. He was able to download the Blaster patch on the next go round and fix it. But it had been 1 or 2 minutes at most before the infection took place.

Most attacks are aimed at Microsoft Windows or specific Windows software from Microsoft, such as Outlook or Internet Information Server. Microsoft has patched known holes, but many Windows users don't download and install these security patches, leaving their computers open to further exploitation and infection.

TIP If you're using Windows, stop reading right now and use Windows Update (access it in Start > All Programs > Windows Update) to install all security patches released by Microsoft! If you're a Mac or Unix user, encourage all your Windows-using friends and colleagues to do the same. If everyone would stay current on security patches, most worms would have much less impact.

Although some viruses exist for Macs, the number is a fraction (and a tiny fraction, at that) of those aimed at Windows, which reduces the worry for Macintosh users. Also, since Macs are a much smaller percentage of the overall market, most crackers haven't been particularly interested in breaking into Macs. It's also difficult to force a Mac user to execute an attachment in an email program unintentionally, or to convince a Macintosh email program to execute malicious code attached to an email message, which are two of the primary methods by which Windows viruses spread.

When you combine that lack of interest with the architectural accidents that made Mac OS 9 and earlier highly secure, you can see why Macs haven't suffered much from security concerns. That is changing now that Apple uses Unix underneath Mac OS X; although

Unix isn't inherently insecure, it's a more common target for crackers, and security holes in widely used programs like Apache are regularly reported. But Mac OS X and the various flavors of Unix systems seem to resist spreading disease: attacks generally compromise one box at a time. Like Microsoft, Apple regularly releases security updates via the Software Update utility (choose Software Update from the Apple menu); we always recommend installing them, although it can be a good idea to wait a few days for any unforeseen problems to appear and be fixed.

Other attacks use what's called a *denial-of-service (DoS)* approach, where the attacker sends so much data to your computer that it's overwhelmed. DoS attacks don't cause damage, per se, but they prevent normal operation and can be difficult to shut down. They are unfortunately quite frequent.

NOTE A DoS attack once saturated Adam's dedicated Internet connection; only calling his ISP and having it block the offending traffic fixed the problem.

The entire issue of protecting your computer becomes much more complicated when you're roaming. The wireless networks themselves could be untrustworthy (is the Internet café's resident geek probing your system?) or a cracker at the next table could be probing your computer directly. Remember, if someone can monitor your un-encrypted network traffic and steal your passwords, she can often use those passwords to enter your machine while it's still on the network.

A little precaution, such as encrypting your passwords and installing a firewall, goes a long way toward preventing an ocean of pain and suffering.

TIP Always back up your data before you take a laptop on the road—even if you're completely safe from crackers, you may drop and break the computer while going through an airport security check or someone may steal it while you're looking the other way. Everyone loses data at some point, and those with backups suffer the least because of it. For more information, see *Take Control of Mac OS Backups*. http://www.takecontrolbooks.com/backup-macosx.html

Install Anti-Virus Software

There are tens of thousands of viruses that can attack computers running Microsoft Windows, and a few viruses have been reported for computers running the Mac OS and Unix as well. These viruses use a variety of methods of infecting computers, and although many are essentially harmless (perhaps only causing crashes due to poor programming), many others are inherently malevolent, with code that causes them to delete or corrupt files, or even erase your hard disk. Also problematic are macro viruses, which live inside documents written with programs that have some sort of scripting—they most commonly infect Microsoft Office documents due to Office's built-in scripting support.

TIP Avenues for infection include inserting an infected removable-media disk from a friend into your computer, downloading an infected file from the Internet, receiving and opening an infected attachment via email, being attacked over the Internet by an automated program, and more. You can't prevent every possible way you could be infected (although exercising caution is always worthwhile), which is why anti-virus software that restricts access to (and from) and constantly scans your computer is so important on Windows.

Put bluntly, if you're using a Windows computer, you will eventually be infected by a virus unless you run anti-virus software that you keep up-to-date. Since so many Windows viruses appear every month, makers of anti-virus software always provide an automatic update service that ensures that their software can identify and eradicate newly discovered viruses.

Numerous companies have sprung up to provide anti-virus software, but the two most common packages for Windows are Norton AntiVirus from Symantec (http://www.symantec.com/) and McAfee's VirusScan (http://www.mcafee.com/). Most anti-virus packages are fairly comparable in terms of basic functions, so choose among them based on price, usability, support, and other features. We don't care which you choose, just make sure you run some form of anti-virus software and keep it up to date.

Windows XP Service Pack 2 will warn you if you happen to disable anti-virus software, which is a nice method of ensuring consistent safety.

NOTE Honestly, we don't run anti-virus software on our Macs. Since we maintain nightly backups and avoid common virus vectors like pirated software and random CDs from people we don't know, the anti-virus software hasn't been worth the effort in recent years.

We would change our tune if someone discovered a way to execute attachments inside a popular Mac OS X email client that would also redistribute email with the same attachment to others. When that happens, watch us run for an anti-virus download.

Assign Private Addresses for Passive Protection

Running NAT (Network Address Translation) on your gateway eliminates many types of break-ins because NAT addresses are typically private—restricted to the local network—and thus unreachable from the outside world. Whenever a computer with a private address on the local network requests a connection with another machine on the Internet, the NAT gateway rewrites the request so it appears to have come from the NAT gateway, which must have a publicly reachable IP address.

Most of the time, you use NAT because your ISP has assigned you only a single IP address, and NAT enables you to share that among a number of computers on your local network. Thus, the protection afforded by NAT is more of a side effect than its primary function.

If someone tries to attack a network protected by NAT, only the gateway is exposed. A gateway may have some vulnerability, but gateways are typically more capable than computers of resisting attacks because their software is so simple and they don't have many ports open—possibly none at all. Because gateways don't do that much, it's hard to hijack them. Some gateways monitor and even log these attacks for later forensic analysis.

Some people call NAT a "passive firewall," and many manufacturers that advertise gateways with firewalls are really offering only NAT. However, even NAT offers fairly significant protection.

TIP If you want to provide access to a computer on your internal network through a NAT gateway, you can open up a specific port, a process called *pass-through, port forwarding,* or *port mapping*. In essence, you're saying, "All traffic to my single NAT-protected IP address on port 25 should be directed to this computer on my network instead of being ignored by the gateway."

From a security standpoint, opening up specific ports makes more sense than a similar feature, called *DMZ Host,* in which all external traffic is directed through the gateway to a particular local computer. Using a DMZ Host is like using port mapping with all ports; it's easier to set up, but the DMZ host loses all protection from the NAT gateway.

TIP Be aware that NAT doesn't protect you from other users on the same network, such as in a coffeehouse or hotel.

Enable an Active Firewall

An active firewall monitors all data entering and leaving a computer or network. Firewall software can be installed on individual computers or on a network gateway or router. Active firewalls examine inbound and outbound data and allow particular bits (and sometimes alert you) if blocked data matches certain criteria. Inside your network, using a firewall so your network's services are open only to local computers is a fine way to discourage ne'er-do-wells from wreaking havoc.

In an active firewall, you can choose to block or pass only certain protocols, only connections that use specific port numbers, IP addresses, or only specific users. In larger networks, you can combine user authentication with a firewall to ensure that only certain people can carry out certain tasks on the network.

More advanced firewall software identifies patterns of data, and when it recognizes an attack pattern in progress, locks out the IP address the data is coming from, and optionally alerts you. Extremely expensive network firewall hardware can recognize thousands of these attack patterns.

Many firewalls also let you set access rules that vary by day of week and time of day. Thus, when you're paying attention to the network, it can operate at a lower level of security. This makes it easier to perform routine tasks that otherwise might be tedious with the firewall in place.

Practically every wireless gateway we've looked at claims to include a built-in firewall, although they generally just mean that they use NAT, not that they have active firewall capabilities. Refer to your manual for details on how to configure your gateway's firewall.

If you're roaming, or want more granular control, you use personal firewall software on individual computers:

- Windows users could try Windows XP's built-in firewall (which we discuss ahead in Enable the Windows XP firewall), but we recommend the more full-featured ZoneAlarm Pro, a powerful but easy-to-use package that's cheap and well supported; there's an awfully good free version from Zone Labs, too. http://www.zonelabs.com/
- Under Mac OS X, Adam uses the built-in firewall (see Enable the Mac OS X firewall) or Sustainable Softworks' IPNetSentryX. http://www.sustworks.com/site/prod_sentryx_overview.html
- On the Mac, Glenn swears by Intego's NetBarrier X3. http://www.intego.com/netbarrier/
- Under Mac OS 9, Adam likes Sustainable Softworks' IPNetSentry. http://www.sustworks.com/site/prod_ipns_overview.html

NOTE Zone Labs offers a version of ZoneAlarm that works hand-in-hand with some Linksys gateways. See http://www.linksys.com/ for details.

TIP When configuring a firewall, the standard approach is to deny all inbound access, allow all outbound traffic (along with incoming responses to that outbound traffic), and then open specific holes in the firewall. That way, it's much easier to figure out what's happening in an attack, since the set of possible ways through the firewall is small. The only downside is that you must spend time determining which ports to open for unusual programs.

TIP If you use EMC Dantz's Retrospect backup program or Netopia's Timbuktu Pro remote control program, you might go crazy trouble-shooting connection problems with gaining access to certain remote machines, which are usually caused by the firewall being on. Read the FAQs at http://www.dantz.com/ and http://www.netopia.com/ on which ports to open.

Enable the Mac OS X Firewall

In Mac OS X, enabling the firewall is extremely easy:

1. Open System Preferences, and click Sharing to open the Sharing preference pane.
2. Click the Firewall tab, and click Start.
3. Select the checkbox next to any services that need outside access.
4. Click New or Edit to modify the services listed if necessary.

Enable the Windows XP firewall

Commercial firewall software may give you more options and a better interface, but the built-in firewall software in Windows XP will do the job.

NOTE Windows XP Service Pack 2 enables the Windows firewall by default, which is a wise decision. It also warns you if and when there's no firewall protection enabled—such as if you turn off the firewall temporarily and forget to turn it back on.

In Windows XP, follow these directions to set up a firewall:

1. Open Control Panel, and then double-click Network Connections.
2. Select the connection you want to secure (you can repeat this for multiple connections).
3. In the left pane, click Change Settings of This Connection under the Network Tasks area.
4. Click the Advanced tab.
5. Check Protect My Computer and Network by Limiting or Preventing Access to This Computer from the Internet.

Secure Small Office Wi-Fi

Small businesses are the backbone of the U.S. economy, and are an even bigger driver of industry worldwide. Small businesses can use the Internet to appear on even turf with multi-national organizations. But until recently, they couldn't match the information technology (IT) infrastructure of big organizations.

That's changed. A lot of technology that was available only in server hardware or software that cost thousands or even tens of thousands of dollars and was out of reach for a business of even 50 to 100 people has now been scaled down and made affordable.

This change lets offices of 5 to 100 people increase the security of their in-house Wi-Fi networks and the security of their mobile users using wireless networks in cafés, hotels, and at home, all without breaking the bank. In some cases, a small office of even one person can take advantage of these tools.

Small offices tend to love Wi-Fi because it reduces the cost of running a network. Ethernet installation can be expensive if done right: $50 to $200 or more per outlet and then you end up with a maze of wires that must be cross-connected using switches. Wi-Fi means that employees who don't have the heavy bandwidth needs of graphic designers or photo editors can work wherever they like. Even putting a Wi-Fi card into a desktop computer makes sense with the latest wireless network speeds. So why don't more small offices use Wi-Fi?

In a word, security. Many small offices have no dedicated IT personnel, and thus have been leery of installing a wireless network that could expose confidential business or customer information. The solutions for closing security holes that might work for homes or individuals aren't necessarily appropriate for 5, 10, or 100 users.

In this section, we offer some simple and cost-effective suggestions that won't send you scuttling to the classifieds to hire an expensive staffer. Instead, you might be able to set up a secure, small-office wireless network by yourself, or at least spend only a few hours with a consultant.

You can secure your local Wi-Fi network against snoopers and unauthorized access in three ways, and we look at each of these in this section:

- **Shared key** (discussed just ahead): Through a single WPA encryption key that is shared among all users on the network. (WEP is too weak.)
- **Wi-Fi login:** Via an authentication system that provides each user with a unique user name and password and then assigns them a unique network encryption key each time they log in.
- **VPN:** Through a virtual private network (VPN) server that encrypts all data that passes over exposed parts of the Wi-Fi network. VPNs are useful both for protecting wired and wireless traffic on your local network and for protecting traffic from mobile users on the road.

Secure Your Network via a Shared Key

With a shared key, each user on the Wi-Fi network has the same encryption key, which means that each user can—if they run special software—see all the data that every other user on the network is passing back and forth.

For a small office, this is unimportant: your main goal is to keep unauthorized users from gaining access to your network, not to keep one network user from viewing another's traffic. (If that latter issue is important, consider application-level encryption so that SSL or IPsec is used to encrypt data as it passes between, say, a file server and a user; or use an authentication system described next.)

Although there are many provisos, if you rely on a shared key, you should follow these principles.

- **Use WPA or WPA2's new keys:** Small offices shouldn't expose themselves to risk by using WEP, a broken encryption method (see Avoid WEP encryption for more on this issue). Rely on the TKIP key type that's found in WPA or the even stronger AES-CCMP key type that's part of WPA2.

NOTE Older equipment may not support WPA, so you may have to replace it to have the appropriate level of security. (Very few gateways let you mix WEP and WPA, and it's inadvisable.) Fortunately, the newest and most sophisticated gear is inexpensive, making the decision to standardize on WPA-compatible equipment easier.

- **Choose a strong key:** TKIP suffers from a weakness if you choose a short key comprised of words found in a dictionary. Choose long TKIP keys that mix letters, numbers, and punctuation of at least 20 characters.
- **Don't let users see the keys:** In most operating system software, a network administrator or other trusted employee can type in the shared key. This shared key can't be seen by users who lack administrator permissions. Specialized software might let them retrieve it, but only a determined internal hacker would try that.

NOTE We're not telling you to distrust your employees, but if they all know the encryption key then they could accidentally reveal the key, or a disgruntled ex-employee could lurk outside the office and gain access. With a known, shared key, you should change it regularly or at least whenever an employees leaves or is fired. Whenever you change the shared key, every computer on the network must have its key changed at the same time to retain access.

Secure Your Network with Wi-Fi Login

There's an easier, but more complicated, way to provide access to a Wi-Fi network for employees, without managing a shared key for the entire network. A system known as 802.1X, another IEEE standard, works with user accounts and WEP, WPA, and WPA2 to provide a unique encryption key for each user on each network login.

802.1X also adds accountability: you know which users are on the network and can specifically lock out particular users. More advanced systems let you set policies that might only allow certain users during working hours or provide guest credentials for 30 days.

NOTE The 802.1 task group is one of the overarching networking groups in the IEEE, and the capital X in 802.1X indicates that it is the full standard. 802.1X works with Ethernet as well as Wi-Fi, but requires more advanced Ethernet switches than are found in small offices to have the same benefit of locking out access.

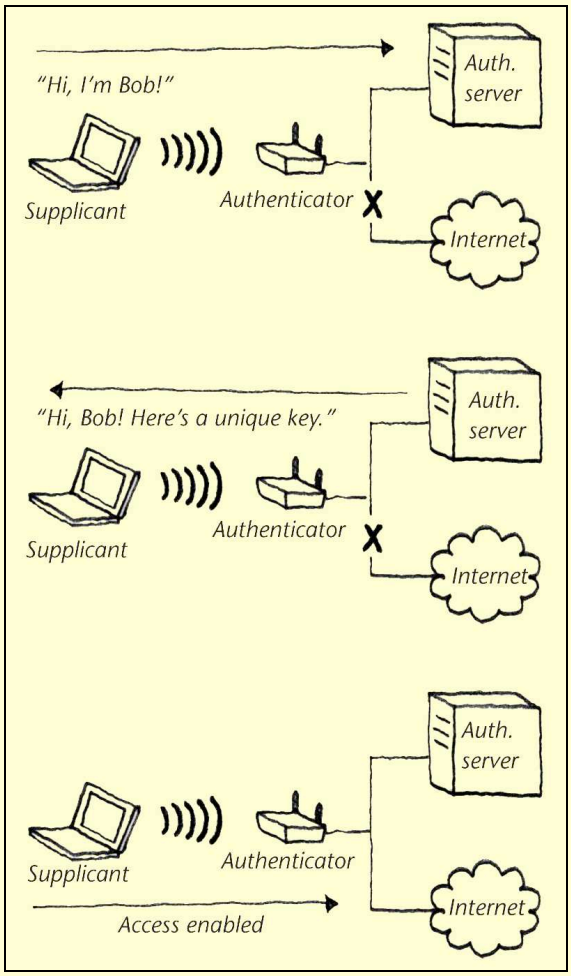
We first give you some background in the underlying principles of this kind of authentication, and then we provide practical advice on implementing it without great expense.

Understand 802.1X

The 802.1X protocol is essentially a way of putting a gatekeeper in front of a network. The gatekeeper prevents network access until a client proves itself worthy by providing credentials that can range from a simple user name and password up to fingerprint or hand geometry, confirmed by a biometric control system.

802.1X defines three roles: a client, which is called a *supplicant*; an access point, which acts as an *authenticator* or gatekeeper; and a user database server or *authentication server*, which confirms a user's identity (**Figure 12**).

FIGURE 12



1. First, a supplicant, having associated with an access point, sends its login credentials (like a user name and password) to the authenticator, which passes them to the authentication server. The access point doesn't give the supplicant Internet access at all.
2. Next, the authentication server confirms the supplicant's identity and responds, again via the authenticator, providing the supplicant with a unique encryption key for using the local network.
3. Finally, with the encryption key in hand and its credentials confirmed, the supplicant can pass traffic through the authenticator out to the Internet (or the local network, as the case may be).

When a user (supplicant) wants to join the network, she uses an 802.1X client to log in (**Figure 13**). (This client is included or available for all current operating systems and handheld organizers; see

the note below.) The authenticator (access point) receives a request for a login from the supplicant. The supplicant can't access any network resources at all except a single network port devoted to handling 802.1X logins.

FIGURE 13

An 802.1X login, shown here in Mac OS X's Internet Connect application; it looks the same in Panther and Tiger.

The authenticator sends the user's credentials to the authentication server, often the same one used for regular network logins to access file servers and other network resources (Step 2 in **Figure 12**). The authentication server tells the authenticator that the login is valid, and the authenticator opens up network access to the supplicant (Step 3 in **Figure 12**).

NOTE Microsoft's Wireless Network Connection tool includes 802.1X support as does the Internet Connect application in Mac OS X 10.3 Panther and 10.4 Tiger, while companies like Meetinghouse and Funk provide commercial software that works on many platforms, including many versions of Windows, Unix, Linux, and Mac OS, along with handheld devices from Zaurus and Palm or that use Pocket PC or Windows Mobile operating systems. Their client software costs about $40 per computer for a single copy and less in bulk.

The 802.1X transaction has already kept the network safe from those who don't have passwords. But it gets better. Once the user has been verified, the authentication server can provide a unique WEP or WPA

encryption key to that specific client. This way, each client on the network can have its own key, making it possible to maintain link security even with WEP.

To overcome WEP's weaknesses, the authenticator can automatically provide a new WEP key on any terms allowed by the software that manages the keys: after a certain number of packets or period of time. If WPA isn't a feasible option, changing WEP keys every few minutes provides enough security for interim purposes. On WPA networks, the interval between key rotations could be quite long without compromising security.

The only known flaw in 802.1X is that the transaction that results in the network connection isn't encrypted. The user name is typically sent in the clear, and the password, while scrambled, can be extracted in exactly that form and potentially replayed later.

Various companies have proposed ways of taking the authentication messages—which use EAP (Extensible Authentication Protocol), a relative of PPP—and encrypting them inside an SSL-like tunnel (see Encrypt Chunks of Data with SSL). The most popular flavors include:

- **EAP-TLS** (EAP Transport Layer Security, a synonym for SSL): This technique requires that you install unique digital certificates on every computer on a Wi-Fi network, which is worthwhile only in situations that require high security.
- **PEAP** (Protected EAP): Functionally equivalent to EAP-TTLS, PEAP is built into Windows XP/2000 and thus has much wider support. PEAP has more or less won the market. Confusingly, Cisco supports both this standard version of PEAP and offers its own version of PEAP that's incompatible with the widely used version. Cisco's non-standard version is found almost exclusively on networks that rely entirely on Cisco hardware and software.
- **EAP-TTLS** (EAP Tunneling Transport Layer Security): This method is functionally equivalent to PEAP, but less widely supported.

NOTE Cisco has long had its own flavor of EAP with encryption called Lightweight EAP or LEAP. LEAP is completely broken and can be cracked easily. Cisco now prefers PEAP—whether Cisco's own or the more widely supported version—but Cisco released a LEAP upgrade known as EAP-FAST (Flexible Authentication via Secure Tunneling). EAP-FAST has a similar problem in its least-secure mode. In its most secure flavor, it's no better than more widely supported PEAP and EAP-TTLS. Cisco released EAP-FAST as a simple migration path for LEAP users as it involves fewer network changes.

Add 802.1X to your network

The complexities of 802.1X can be hidden from view entirely because many wireless gateways, including most inexpensive ones from Linksys and others, can work as an 802.1X authenticator.

Users simply enter the IP address, or host and domain name of the authentication server, and a shared key that's kept between the access point and the server, and they're all set. This set of values is usually listed as RADIUS settings; RADIUS (it no longer stands for anything) is a type of user account management server.

TIP The firmware on early MIMO (multiple-in, multiple-out) gateways lacked 802.1X support. That's been rectified in newer releases; upgrade your router if necessary.

At one point, all 802.1X servers cost thousands of dollars whether they were standalone products or part of a RADIUS system. Now, there are five inexpensive options that provide full 802.1X benefits to a small office: the first two are software products you install in your office; the next three are hosted solutions that run remote 802.1X servers on your behalf over the Internet:

- **Elektron:** Elektron is a WPA Enterprise server that runs as a separate program on an existing computer. It can tie into Windows Active Directory or Mac OS X (plain or Server) user directories. Elektron can authenticate remote access points over the Internet as well as local Wi-Fi gateways on the same network. It runs under Mac OS X 10.2 or later and Windows XP, 2000, and 2003. Any standard 802.1X client works with Elektron using PEAP or EAP-TTLS—both versions are available simultaneously. The enterprise

version supports more advanced directory integration among other features. See Appendix A: Use WPA Enterprise for details on setting up Elektron. (Corriente Networks; $299 basic, $749 enterprise, unlimited users; http://www.corriente.net/)

- **LucidLink:** LucidLink requires a special software client that works only on Windows 2000 or XP Service Pack 1 or later. The server software runs on Windows 2000, Server 2003, and XP. Users don't need to manage a user name or password; their computers are given permission to access the network by someone who doesn't need any technical skills. (Interlink; 1–3 users: free, 4–10 $449, bundles for larger numbers of users; http://www.lucidlink.com/)

- **SecureMyWiFi:** WiTopia is a plain outsourced WPA Enterprise server administered through a Web site. It's the cheapest option by far ranging from free for small offices to a fraction of the price of competing services per user. Works with any standard 802.1X client that supports PEAP or EAP-TTLS (only one method at a time). (WiTopia; 1–5 users and 1 access point: free for one year, additional groups of 5 users: $5/group/year, additional access points: $10/access point/year; http://witopia.net/aboutsecuremy.html)

- **WSC Guard:** WSC Guard is hosted over the Internet. A Web site is used to add and change user information. The client software supports Windows 98, Me, XP, and 2000. A failover server is available that can run on a Windows machine in your office; if your Internet link goes down, this in-house server allows users to join the network. (Wireless Security Corporation; 1–4 users: $4.95/month/user, 5 or more: $3.99/month/user; http://www.wirelesssecuritycorp.com/)

- **BoxedWireless:** BoxedWireless works with any standard 802.1X client that supports EAP-TLS or Microsoft's flavor of PEAP. This is the only firm that can provide individual certificates, making it affordable for organizations that want the security that goes along with this method but doesn't want to build their own public-key infrastructure. Accounts are configured via a Web site. (BoxedWireless; 1–10 users: €19 (US$23)/month, 11–20 users: €39 (US$47)/month, 21–50 users: €79 (US$97)/month; http://secure.boxedwireless.com/)

Secure Your Network with a VPN

We describe the nature and utility of VPNs in Encrypt All Data with a VPN. In brief, a VPN connection securely encrypts all the data entering and leaving a computer. Until recently, you would have had to spend many thousands of dollars and pay for real IT expertise to purchase, configure, and maintain a VPN server or service.

Virtual private networks can improve small office network security in two basic ways:

- A VPN is essential for users who leave the security of your local network. Using a VPN is our primary recommendation for securing a long-distance connection over any network—not just from hot spots, but also from a hotel's wired broadband or from guest connections on other organizations' networks.
- You can implement a VPN for wireless users connecting directly to your local network. Although this may seem like (and, in fact, may be) overkill, it provides the maximum security possible, particularly when coupled with appropriate use of WPA2.

NOTE If you use a VPN to secure wireless users connecting directly to your local network, you must take two steps:

- For local users on your local network to see each other's computers and network servers when using a VPN, you must host your own VPN server software or hardware, as described in the next two sections.
- You must configure your access point to ban all inbound network traffic except that aimed at VPN ports. (Allow port 1723 for TCP and UDP for PPTP, and port 1701 for TCP and UDP for IPsec over L2TP. SSL doesn't require a special port, but uses 443 just like a secure Web site request.)

This second step means that a VPN client can connect via the access point and to the VPN server, but that all other inbound connections will be rejected. Because local clients will tunnel all traffic over that VPN connection, they have full access to your network.

Fortunately, recent changes have revolutionized the VPN world: you can now find "free" (or built-in, at least) VPN client software in Windows XP and Mac OS X 10.3/10.4 for PPTP- and IPsec-based

VPNs (Mac OS X 10.2 supports just PPTP), and you can purchase a variety of VPN servers (hardware, software, and hosted services) without spending much on equipment or monthly service charges.

Table 3 summarizes the options for creating a VPN server, and we discuss each option in turn, later in this section.

Table 3: VPN Server Options for Small Offices

Option	Pros	Cons
Run free server software on your own server	Free; you can configure exactly what you want	Requires more configuration and support
Run commercial server software on your own server (Mac OS X Server, Windows Server 2003, etc.)	You can configure exactly what you want; potentially easier configuration than with free tools	Requires more configuration and support; often costs hundreds or even thousands of dollars
Run a special device that acts dedicated VPN server	New model from Buffalo costs only $160, doubles as Wi-Fi gateway	Limited protocol support
Subscribe to a VPN service on the Internet	Requires little configuration or support	Monthly or yearly charge

TIP Businesses with more resources or specific needs might want to hire a corporate VPN service provider. Many companies provide such services, but prices are all over the place and some providers may require installing purchased or leased hardware on your network. Check out http://findvpn.com/providers/ for a lengthy list of VPN service providers.

Choose a software VPN server

Most large organizations use dedicated hardware to support hundreds, or even thousands, of VPN connections; these hardware devices rely on special chips to handle the massive computation necessary to encrypt and decrypt all network traffic.

But if you support just dozens of simultaneous users, you can use software installed on a server that might already also act as a Web

or email server. If you must combine VPN and other services, monitor performance carefully to ensure that you're not being penny-wise and pound-foolish by destroying your Web performance to run your VPN.

It's not inconceivable that you might have the time, money, and know-how to run your own software-based VPN server, but make sure you consider the ongoing support time and effort. Here are some likely choices:

- **OpenVPN:** This free, open-source project offers client and server components that use SSL as their basis of starting a VPN tunnel, with any of a large number of encryption algorithms up to the highest level of publicly available government-grade encryption. It's available for many platforms, often in an installer form. http://openvpn.net/
- **Windows Server 2003:** This high-end Microsoft product, which costs thousands of dollars, might already be installed on your network if you're running a Windows shop and need the kinds of services it provides. Windows Server 2003 includes full support for PPTP and IPsec-over-L2TP VPNs. Its advantage is that there are many Microsoft certified technicians who can consult on configuring and maintaining it. The downside, of course, is cost: a 25-user version with all the trimmings costs $4000. http://www.microsoft.com/technet/prodtechnol/windowsserver2003/deploy/confeat/rmotevpn.asp
- **Mac OS X Server 10.3/10.4:** Apple revised this server software in version 10.3 to include PPTP and IPsec-over-L2TP VPN servers. The software runs only on Macintosh hardware, and includes a host of other network services, just like Windows Server 2003. The difference? Apple's server allows unlimited users and costs just $999; a 10-simultaneous-file-sharing-user version is $499 (VPN and other users are unlimited). Apple's Xserve rack-mounted system includes an unlimited version of Mac OS X Server in the price, which is an even better deal, given that Xserves start at $2999, just $2000 more than the software alone. Or buy a $1000 tricked-out Mac mini coupled with a $499 server. http://www.apple.com/server/macosx/

- **Guardian Digital Secure VPN Server Suite (Linux):** This VPN server is relatively inexpensive because it's a package of open-source software that's been put together for ease of use. There's no limit to the number of users with the cheapest version at $595: you pay more for additional phone, email, and Web support. http://store.guardiandigital.com/html/eng/products/software/vpn_overview.shtml

Choose VPN hardware

When people talk about VPN hardware, they usually refer to systems like the Securepoint Firewall & VPN Server Appliance (http://www.securepoint.cc/), which starts at thousands of dollars for a handful of users.

But we're excited about a new generation of VPN hardware aimed at the small office that costs a fraction of that. The first device we've seen that combines low price and high ease of use is Buffalo's incredibly long-named 125 High-Speed Mode Wireless Secure Remote Gateway (http://www.buffalotech.com/products/product-detail.php?productid=88).

The Buffalo device supports a number of techniques to speed up Wi-Fi when used with compatible devices, but its real charm is that it has a PPTP VPN server built right in that can handle up to 100 user accounts. It's often discounted to as little as $160.

The Buffalo gateway also offers another unique feature: users logged in via the VPN can exchange files with each other via supplied software no matter where they are. This feature makes it easy for two users working at the same remote location to swap files without using an insecure medium.

TIP Because the Buffalo gateway supports only PPTP, make sure to choose a long, complex password to avoid a well-known crack that makes shorter PPTP passwords susceptible to exposure.

Choose a VPN service

If the VPN options listed above made your pocketbook ache or your eyes glaze over, consider another alternative for roaming users: subscribing to a service that offers VPN connections from wherever the user is to a secure network operations center (NOC) elsewhere on the Internet.

Typically, the least secure link in any connection is the local network: the sniffed or penetrated Wi-Fi or wired network over which traffic proceeds unencrypted out to the Internet connection. Once traffic is on the broader Internet, it's much less likely that any snoop would be able to intercept it—in essence, your traffic becomes more secure once it leaves the local network you're connected to. By creating a secure client-to-NOC connection, you eliminate many of the most common places that people could sniff or intercept network traffic.

NOTE These VPN services certainly increase your security level significantly, but if you need complete security back to your local network, you need a software- or hardware-based VPN server running on your local network. Otherwise, it's possible that your traffic could be snooped after it leaves the VPN service's NOC.

Several companies offer VPN services for hire that are specifically designed for hot spot users. We've tested and have experience with three of them:

- **HotSpotVPN:** This service offers two flavors of security:
  - HotSpotVPN-1 is a standard PPTP service that used to be the company's flagship offering. It is available for $3.88, $5.88, and $6.88 in 1-, 3-, or 7-day increments.
  - The more robust SSL-based HotSpotVPN2 service uses the OpenVPN software for almost every platform, with monthly charges based on encryption algorithm: 128-bit Blowfish, $10.88 per month; 192-bit AES, $11.88 per month; and 256-bit AES, $13.88 per month. The HotSpotVPN2 service includes a free HotSpotVPN-1 subscription.
    http://www.hotspotvpn.com/
- **WiTopia's PersonalVPN:** This $39.50-per-year service uses OpenVPN software for an SSL connection and provides it at a

much lower cost than HotSpotVPN does. But it offers just the 128-bit Blowfish cipher.
http://www.witopia.net/

- **PublicVPN.com:** This service costs $5.95 per month or $59.95 per year, and it deploys IPsec VPN tunnels and works with any standard IPsec client.
http://www.publicvpn.com/

Perform a Security Audit

We've spent the ebook so far helping you figure out how to increase the security of your wireless network. Now it's time to determine how successful you've been with a security audit.

This is the basic information you need to break into your own network. If you can break in fairly easily with readily available tools, that's a good indication that what you've done won't survive a serious attack from a knowledgeable cracker.

Warning! Before we go further, we need to make two important points.

First, we are not network crackers, nor do we play crackers on TV. As such, we can't guarantee that real crackers won't employ even more capable tools or techniques.

Second, some of the tools and techniques we do discuss are possibly illegal—depending on your state and country laws—when applied against networks that you don't own or have permission to access. Use our advice in the spirit it is intended—as a way of giving you the peace-of-mind that your wireless network is sufficiently secure—and don't turn to the dark side.

Before we get started, though, you need to collect some software.

Assemble Your Tools

The first step in any security audit is putting together a tool kit of the necessary software. Everything we discuss here is free; most of it is open source, which means that it's equally as available to the bad guys as it is to you.

Although we've listed some of the more common software, the field changes constantly, and there are oodles of additional tools. Check out these lists for more:

- http://www.cromwell-intl.com/security/monitoring.html
- http://www.networkintrusion.co.uk/wireless.htm

Wireless client software

Most types of wireless client software automatically detect open Wi-Fi networks and present you with a list of available networks. For instance, in Mac OS X, simply look in the AirPort status menu, on the menu bar, to see available networks. In Windows, right-click the Wireless Network Connection icon in the System Tray and choose View Available Wireless Networks to see a list of available networks.

Because everyone has basic wireless client software, it's worth checking it to make sure you know what people using it see when they're in range of your network.

Wireless stumblers

An advanced wireless user will likely have a *stumbler program,* which looks for accessible wireless networks, displays those it finds, and presents additional information about each one. Data that a stumbler program can provide includes network name, channel, signal strength, and WEP/WPA encryption status.

For a list of common stumblers, all of which are free, but none of which can detect closed networks, see **Table 4**. Pay special attention to the hardware requirements before downloading, since many work with only certain wireless cards.

TIP You should always have a stumbler in your wireless tool kit, since it can report on signal strength as you move around, which is helpful for finding the best spot for locating access points. They can also be useful for identifying rogue access points.

Table 4: Common Stumbler Utilities			
Stumbler	**Platform**	**URL**	**Notes**
MacStumbler	Mac OS X 10.1 or later	http://www.macstumbler.com/	Supports AirPort and AirPort Extreme
iStumbler	Mac OS X 10.2 or later	http://www.istumbler.com/	Supports AirPort and AirPort Extreme, can also detect Bluetooth devices and Bonjour (Rendezvous) services
ClassicStumbler	Mac OS 9	http://www.alksoft.com/classicstumbler.html	A good use for that old PowerBook
Netstumbler	Windows 2000 or XP	http://www.stumbler.net/	Works with a large variety of wireless cards
MiniStumbler	PocketPC 3.0 or later	http://www.stumbler.net/	Useful for creating a highly portable wireless network scanner

Wireless sniffers

For the most part, sniffers can perform the same tasks as stumblers, but they're more limited in the particular wireless cards that they support because they require lower-level access to the radio technology in the Wi-Fi card. For instance, KisMAC works only with old 802.11b AirPort cards, not the more recent 802.11g AirPort Extreme cards because Apple's chip partner doesn't release information publicly about the chips' inner workings.

In exchange for that limitation, sniffers can detect closed networks (which are hidden to stumblers) and the active probes used by stumblers. In other words, sniffers can detect in-use stumblers.

Because sniffers are used regularly in security audits, some have gained the capability to list connected clients, capture traffic, break passwords, and more. Sniffers are powerful and dangerous tools, and it's essential that you know how they can be used against your network. **Table 5** summarizes well-known sniffers.

WARNING! Sniffers are not the easiest programs to use, and, in particular, the Unix tools are not for the inexperienced.

Table 5: Popular Sniffer Utilities

Sniffer	Platform	URL	Notes
KisMAC	Mac OS X 10.2 or later	http://www.binaervarianz.de/projekte/programmieren/kismac/	Extremely powerful, but limited to AirPort
AirSnort	Unix, Windows XP	http://airsnort.shmoo.com/ http://airsnort.shmoo.com/windows.html	Specialized sniffer designed to capture traffic and break WEP passwords
Kismet	Unix	http://www.kismetwireless.net/	Full-featured sniffer
bsd-airtools	BSD-based Unix	http://www.dachb0den.com/projects/bsd-airtools.html	A collection of Unix tools for sniffing networks, cracking WEP keys, and more

Network analyzers (wired and wireless)

All of the sniffers in **Table 5** are designed to monitor and crack wireless networks. These sniffers can capture data without even associating to a Wi-Fi network, but as long as your network is encrypted (and the key can't be broken), they won't be able to make any sense of the contents of that traffic.

However, there's also a class of software—traffic analyzers—that can capture and analyze wireless and wired network traffic more generally; for such a traffic analyzer to work, it must be running on a computer that is connected to the network in question. **Table 6** lists well-known traffic analyzers.

Table 6: Common Traffic Analyzers

Analyzer	Platform	URL	Notes
EtherPEG	Mac OS 9, Mac OS X	http://www.etherpeg.org/	Proof-of-concept that shows graphics traversing unencrypted networks
WinDump	Windows	http://www.winpcap.org/windump/	Port of the Unix tcpdump utility
tcpdump	Unix (including Mac OS X)	(Run from the Unix command line)	General utility for capturing traffic; simpler and more primitive than Ethereal, but similar
Ethereal	Unix (including Mac OS X), Windows 98/Me/2000/XP	http://www.ethereal.com/	General sniffer for any network type, not just wireless; similar to tcpdump, but provides a graphical interface
ntop	Unix (including Mac OS X), Windows	http://www.ntop.org/	Provides a Web interface for watching and capturing traffic
ettercap	Unix (including Mac OS X), Windows	http://ettercap.sourceforge.net/	Specialized analyzer for capturing passwords; requires you to compile source code

Check Your Network Name/Admin Password

Stop! Do not pass Go. Do not collect $200! Instead, log in to your wireless gateway and verify that you have changed the network name to be something other than the default and that you've assigned a proper admin password to replace the stock password that ships with most gateways. (See Change Network Name and Admin Password.)

Many people fail to change these defaults, which means that it's trivial for anyone who knows enough to look for a network called **`linksys`** to try to log in with the password **`admin`**. (And no, we haven't given away any great secrets here; this stuff is common knowledge among anyone who uses wireless networking gear much—in fact, common network names with their default passwords

are embedded in some wireless snooping software.) Also, failing to change your default network name can significantly reduce the security even of WPA, since it creates its keys in part by using the network name.

Check Availability

Once you've verified that you haven't left the door to your wireless gateway wide open, the next step in your security audit is a tour of the grounds, evaluating whether people will be able to see and attempt to join your network. To accomplish this, walk around the public areas on the perimeter of your network with a laptop.

> **TIP** For your tour, we recommend using a laptop with particularly good reception to simulate the widest possible coverage area. In the Macintosh world, that translates to using a plastic-cased iBook in favor of a metal-cased PowerBook; for Windows, it depends entirely on the wireless card and antenna setup used in a laptop.

Pay special attention to places where a cracker could spend time unnoticed but still in range of your network, since many attacks depend on having sufficiently lengthy access in order to gather a lot of traffic. In a few different locations as you walk around, do the following:

1. **Simulate what a casual passerby will see:** Check your wireless client software to see if your wireless network appears. If your network is open, it will appear; if it's closed, it shouldn't appear.
2. **Run a stumbler:** Slightly more advanced casual users may, instead of relying solely on their operating system to inform them of available networks, run a stumbler program. As such, it's a good idea to do the same, just so you know what such a user will see.
3. **Run a sniffer:** Stumblers won't see your network if it's closed, of course, but as we noted in Don't bother closing your network, anyone determined to break into your network won't bother with a stumbler anyway, but will instead rely on a sniffer that can see closed networks. In a sniffer, pay special attention to how the signal strength of your network drops off as you move further away from your access point. Mapping signal strength in this way enables you to identify places that a cracker might be able to lurk unnoticed while still accessing your network.

WARNING! We almost hate to mention this, since the likelihood of it happening is quite low, but it is possible for a cracker to sit outside the normal range of your network and still access your network with the assistance of a high-gain antenna that can receive the otherwise too-weak signals. In other words, although your tour is entirely worthwhile, the most serious of crackers may be able to access your network from areas you can't secure or watch. This may be partly why, for instance, the Los Alamos National Laboratory in New Mexico has banned Wi-Fi networks altogether.

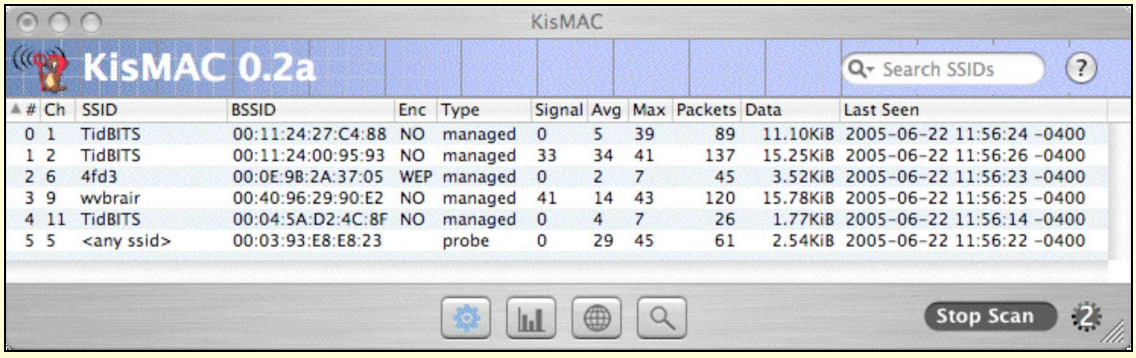
More generally, this translates to the lesson that physical isolation doesn't inherently imply security; for real security, you must employ encryption of some sort, ranging from WPA to a VPN.

Identify Connected Clients

Once you've made your tour, it's time to settle down with a sniffer like KisMAC (http://www.binaervarianz.de/projekte/programmieren/kismac/) and see what a determined attacker can do to break into your network. We'll use KisMAC as an example here, but other sniffers can perform roughly the same tasks.

Initially, KisMAC lists all the access points it can see (**Figure 14**), providing useful information such as the MAC address of the access point (the BSSID column), whether or not encryption is turned on, signal strength, and the amount of traffic.

FIGURE 14

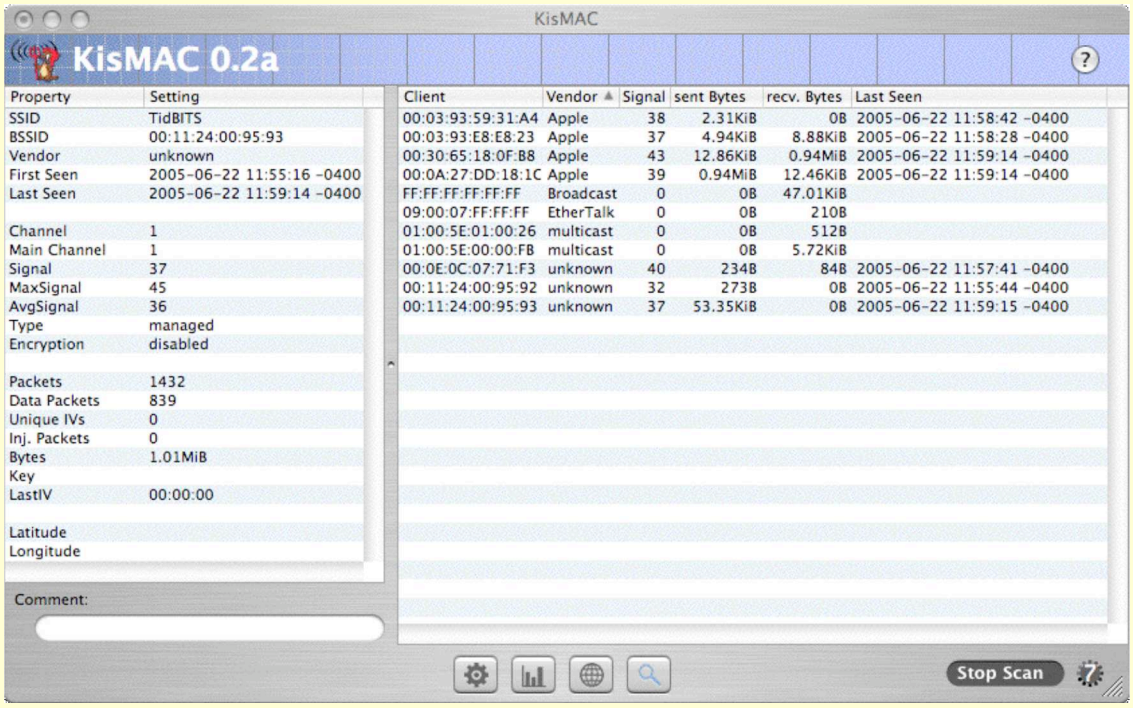


KisMAC shows all wireless networks within reach, including those that are closed (there are multiple TidBITS networks because Adam has an AirPort Extreme Base Station, and AirPort Express Base Station, and a Linksys gateway, all of which share the same name to enable roaming from one to the other).

TIP Notice line 5 in **Figure 14**, the one with an SSID of **`<any ssid>`** and the type **`probe`**. That's actually a copy of iStumbler running on another laptop on the network. Sniffers can't see other sniffers, since they're passive collectors of packets, but they can see stumblers, which send out active probes. Stumblers have plenty of legitimate uses, but if you want to see who's running one, you may be able to find them by watching the signal strength reading as you walk around with a laptop running a sniffer.

Now that you see all the access points in your vicinity, double-click one to drill down and see more information, including a list of connected clients (**Figure 15**).

FIGURE 15



KisMAC showing more details about a particular access point, including the connected clients.

KisMAC provides the MAC address of each client, along with the vendor name in some instances; that can help you identify particular clients. It also includes the signal strength currently enjoyed by each client, along with the amount of data sent and received.

NOTE In Ignore MAC address-based access controls, we noted that limiting access to only computers whose MAC addresses you added to an access list was more trouble than it's worth. As you can see in the client list in **Figure 15**, it's trivial to find out which MAC addresses are allowed on a network. To convince yourself that it's easy enough to change the MAC address to masquerade as a privileged client, read the Wikipedia entry on MAC address (http://en.wikipedia.org/wiki/MAC_address) or do a Google search on **`spoof MAC address`** and scan the top hits.

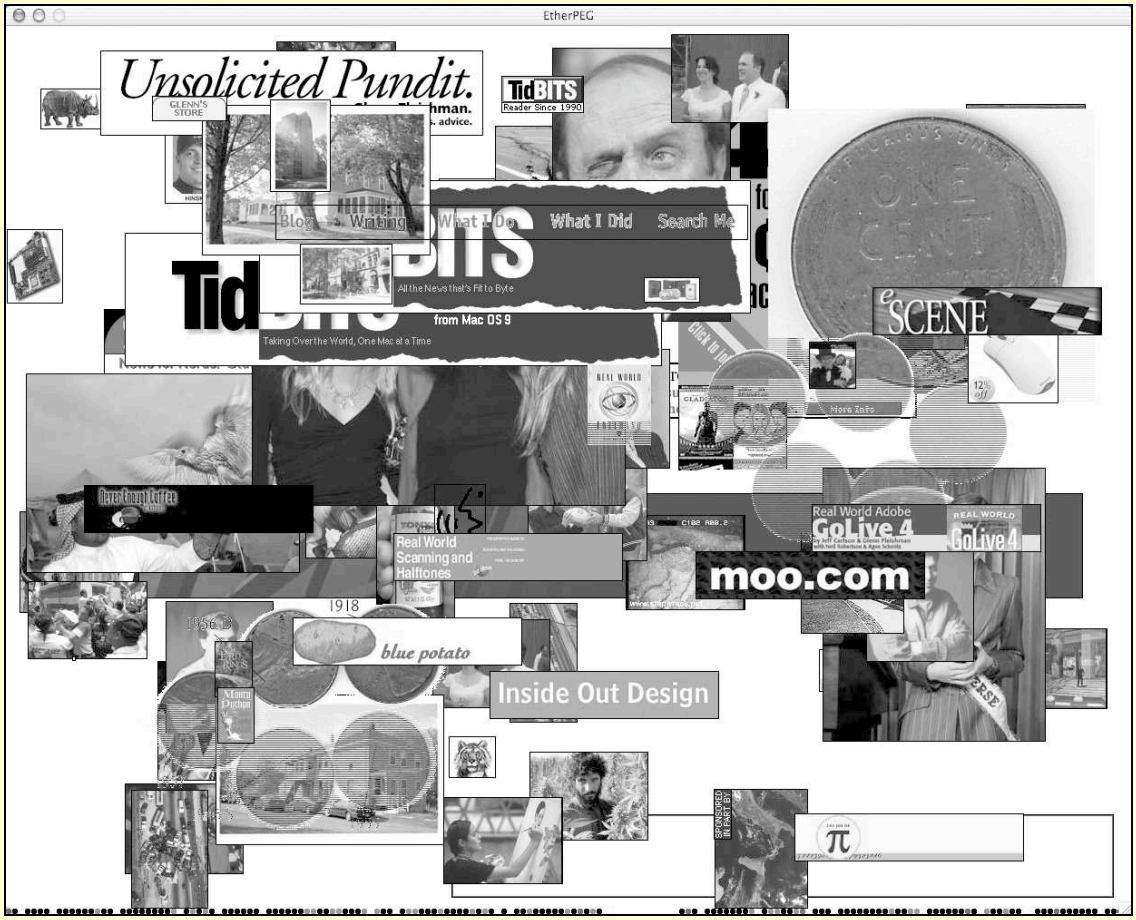
Needless to say, if you're managing a large wireless network, trying to identify which MAC addresses are allowed and which aren't would require a near photographic memory (or clever software). But on a small network with relatively few regular users, it's worth keeping a list of the allowed MAC addresses and comparing that to the list of connected clients every so often.

TIP You don't have to rely on sniffer software to see a list of connected clients; many wireless gateways include the feature in their management software. Apple's AirPort base stations do not, but the free AirPort Management Tools enables you to access the data from an AirPort base station (http://www.apple.com/support/airport/).

Capture Traffic

Now that you see how easy it is to identify clients on your network, both for you and for an attacker, let's capture some traffic. Just for fun, if you're using a Mac, try EtherPEG (http://www.etherpeg.org/), a special proof-of-concept traffic analyzer that displays GIF and JPEG images transmitted over unencrypted wireless networks. It's a bit old at this point, and it was written as a hack at a MacHack conference, so you may not be able to make it run. But if you do, you'll see something like **Figure 16**.

FIGURE 16



EtherPEG gives you a graphical look at the traffic flowing across an unencrypted network.

For a more general look at the traffic that's flowing over your network, turn to another of the traffic analyzers, such as tcpdump, which is included with many Unix distributions. To get a sense of your network's traffic if you can use tcpdump, invoke it at the command line with: **`sudo tcpdump -i en1 -s 0 -A`**

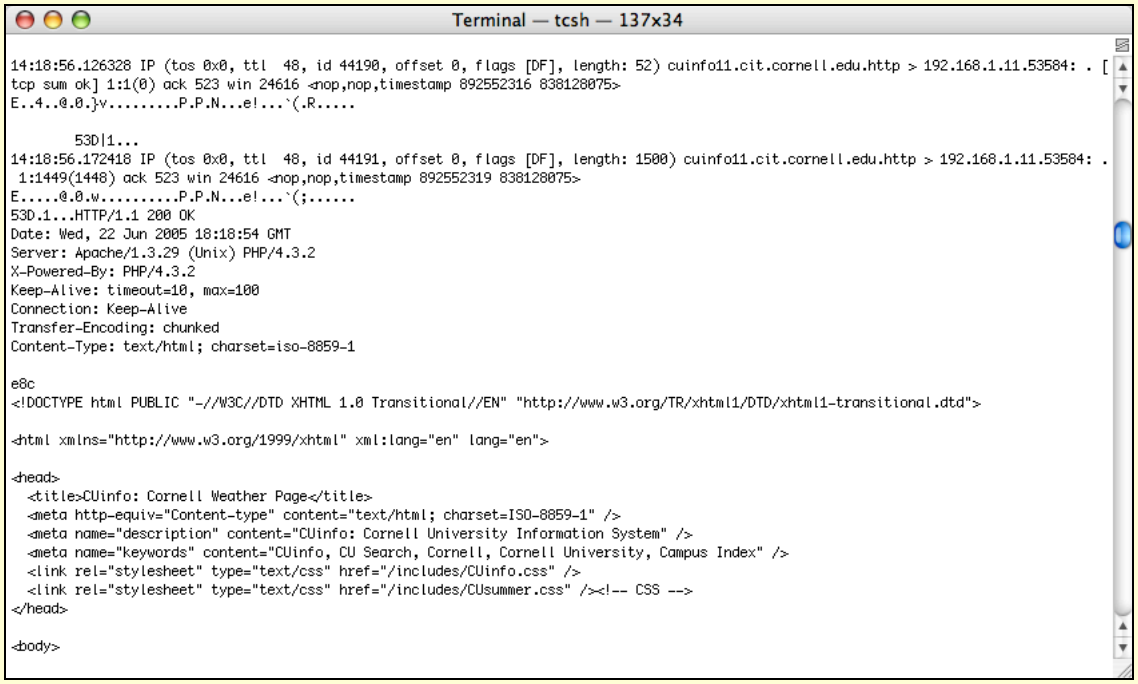
Let us explain the parts of that command:

- You need to use **`sudo`** to run tcpdump with root privileges, and you'll have to enter your administrator password after doing so.
- **`-i en1`** tells tcpdump to use network interface #1, which is normally your wireless connection (en0 is usually wired Ethernet). To check the name of your wireless card's interface, use the **`ifconfig`** command.

- `-s 0` tells tcpdump to read all of each packet (you can also replace the zero with a number like 1500 to capture the first 1500 bytes of each packet).
- `-A` makes tcpdump print each packet in ASCII so it's more human-readable.

When you run tcpdump, you'll see a result that looks roughly like **Figure 17**; press Control-C to stop capturing. The output from tcpdump isn't pretty, but you can extract useful information from it.

FIGURE 17



The screenshot shows data captured by tcpdump; in this case it's the HTML code from the Cornell University Weather Page. But it could have been your customer's credit card numbers, your confidential business plans, or any other data that flows over an accessible network.

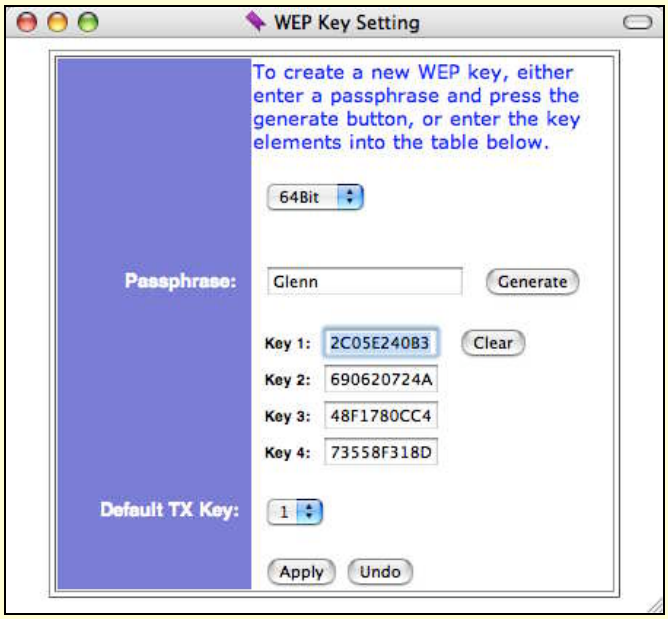
SIDEBAR **CAPTURING PASSWORDS WITH ETTERCAP**

Perhaps the most dangerous utility of all is ettercap, a special traffic analyzer that looks specifically for passwords, such as would be used to log in to a POP account when checking email. To use ettercap, you must be a client on a network, which means that a properly secured and encrypted network is safe; as long as you can keep attackers off your network, they can't use ettercap or another of the traffic analyzers. However, it also means that someone at a coffeehouse hotspot could run ettercap and capture passwords from people who haven't followed the advice in Secure Your Data in Transit.

Crack a Password

For the final part of the security audit, let us show how easy it is to break a WEP key. You won't see us breaking into other people's networks, so we performed a little experiment that we'll walk you through. First, Adam reset his Linksys gateway, turned on 40/64-bit WEP, and generated a key (**Figure 18**).

FIGURE 18



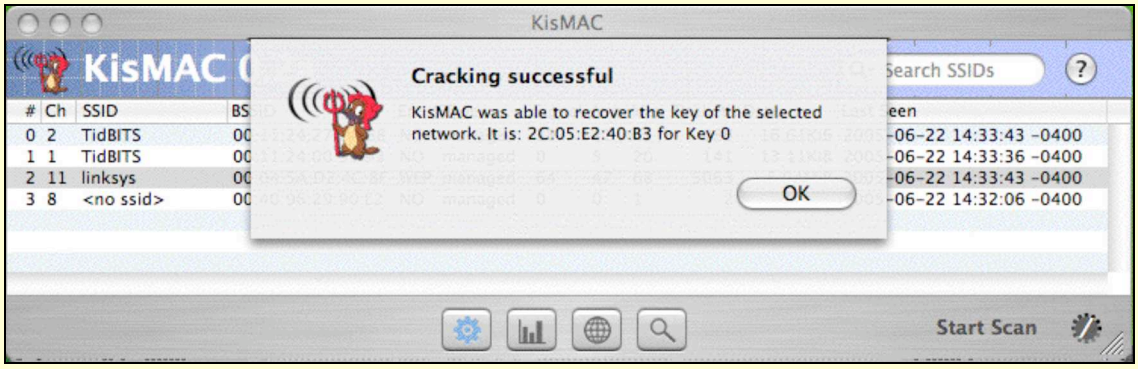
To encrypt his network, Adam created a 40/64-bit WEP key in the Linksys management interface.

Then he connected a laptop to the wireless network created by the Linksys gateway and started copying some files over the network from another machine; the particular attack he used needed some traffic from which to extract the key. But we're not talking about much traffic here—Adam simply copied a few files and loaded a few Web pages.

Then, in KisMAC, he started the attack by choosing Network > Crack > Bruteforce > Newsham 21 bit Attack. This particular exploit relies on a particularly brain-dead method of generating WEP keys from a passphrase (which you'll note in **Figure 18** was `Glenn`): converting ASCII characters to their hexadecimal equivalent to make a WEP key, thus reducing the universe of keys by more than 80 percent.

Access points from Linksys, D-Link, Belkin, and Netgear, and undoubtedly others, are vulnerable to this attack, whereas Apple and 3Com access points use a better algorithm. But since Adam was testing this on a Linksys gateway, the Newsham 21-bit attack extracted the correct WEP key in less than a minute, as you can see in **Figure 19**.

FIGURE 19



KisMAC proudly shows off its cracked password. Compare it against the one in **Figure 18** for verification that it got the right password.

This password-cracking capability is, of course, problematic for two reasons. First, once your password is known, the cracker can connect to your network and do anything that any legitimate user can do. Second and more worryingly, the cracker can also then use ettercap or any other traffic analyzer to capture and analyze all the traffic flowing across your wireless network.

Of course, simply using WPA with a serious password would eliminate from your network the kind of vulnerability that Adam exploited in this example.

Appendix A: Use WPA Enterprise

It's simple to use WPA Enterprise with newer software designed for smaller businesses, and we thought we would run you through an example of installing, configuring, and connecting to Elektron Server from Corriente Networks.

Install a Trial Version

Download the trial version of Elektron Server from http://www.corriente.net/. Install the software on a machine with a fixed IP address and an associated domain name that can be reached from the base stations you want to point at the Elektron Server. The 30-day serial number is on their download page. (If the Elektron Setup Assistant appears automatically at the end of the installation, follow the directions to enter your serial number and create a password. Click Continue to jump ahead to Step 2, just ahead.)

Create a Self-Signed Digital Certificate

Now follow these steps to create a self-signed digital certificate that will secure the connection between Elektron and your WPA-Enterprise clients:

1. From the **`/Applications`** folder, launch Elektron Setup Assistant. Click Continue.
2. **Digital Certificate:** Select Create a New Certificate Hierarchy (Recommended). Click Continue.
3. **Certificate Server Name:** Enter the fully qualified domain name, like elektron.tidbits.com, into the Server Name field. Click Continue.
4. **Certificate Organization and Certificate Location:** In the next two screens, enter your organization, city, state, and country. Click Continue as appropriate. (You can enter any text for these values as long as they fit within the length constraints noted.)
5. **Create Self-Signed Certificate:** Click Continue.
6. **Conclusion:** The final screen allows you to export a certificate that can be installed under Mac OS X, Windows, and (via the text format) other platforms. (You can access these packages and files again within the Elektron Settings application, too.)

This process not only creates a self-signed digital certificate, but also installs all the root information needed for Elektron Server into the system Keychain. When you launch Elektron Settings in the next set of steps, the certificate you just created will be already selected.

NOTE When you connect to a Web server that uses SSL, the server sends you certificate information. Your Web browser automatically checks the validity of the Web server's certificate by consulting your browser's list of signatures for certificate authorities that vouch for most secure Web certificates. If the cryptographic signature of the certificate the Web server gives you matches the particular certificate authority, the server passes inspection.

You can use this kind of certificate with 802.1X or any SSL-based system, but you must pay a yearly fee. You can, instead, employ a *self-signed certificate* that no other authority vouches for. Elektron creates this kind of certificate for you and signs it with their own certificate authority, which you can install on all Wi-Fi client machines. (Technically, because Elektron signed, it is self-signed only to them—but it's not a third-party verified certificate.)

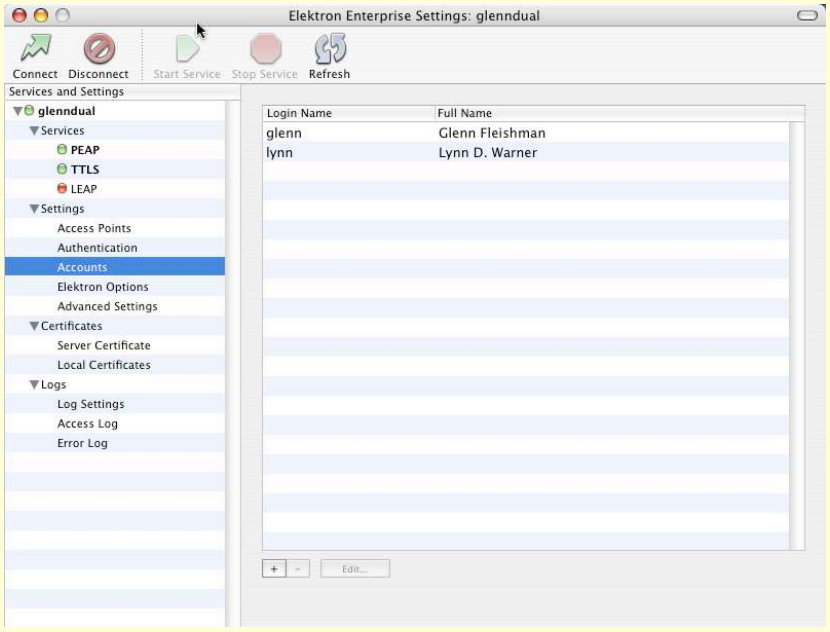
These self-signed certificates and certificate authority validators are stored on your system after you accept them, and they are accessible through Keychain Access, which is located in **`/Applications/Utilities`**.

Configure Elektron Server

Now we need to configure Elektron Server:

1. From the **`/Applications`** folder, launch Elektron Settings.
2. In the Accounts pane, choose Mac OS X Accounts to use accounts already set up on the computer you've installed the software on, or choose Elektron Accounts to create Wi-Fi-only accounts (**Figure 20**). If you choose the latter, make sure to create at least one account to test.

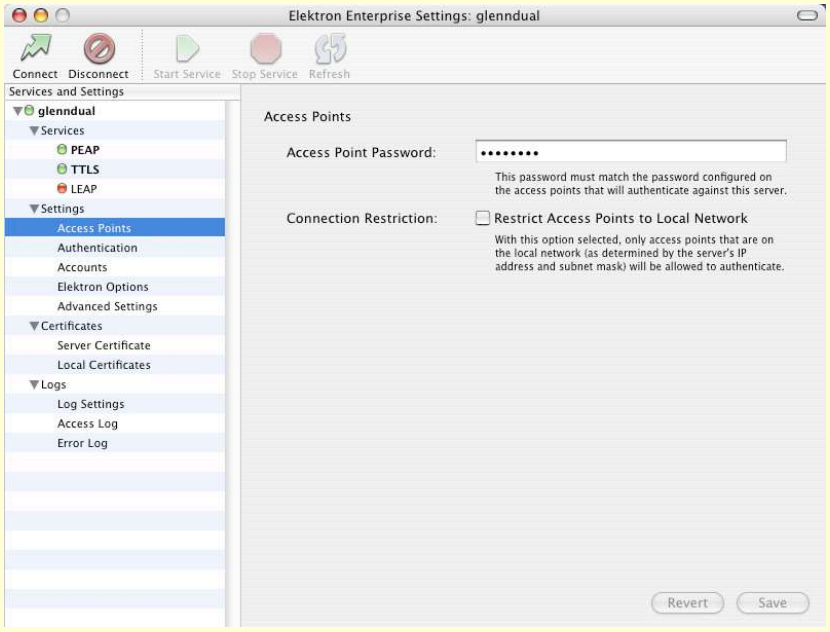
FIGURE 20



Set up user accounts for those who should have access to the Wi-Fi network.

3. Click the Access Points tab. The Access Point Password should be pre-filled from the installation process earlier. If it isn't or you change your mind, enter a passphrase. This passphrase will later also be entered later in the appropriate location in AirPort Admin Utility (**Figure 21**). This is called the *shared secret* in most access points' settings.

FIGURE 21

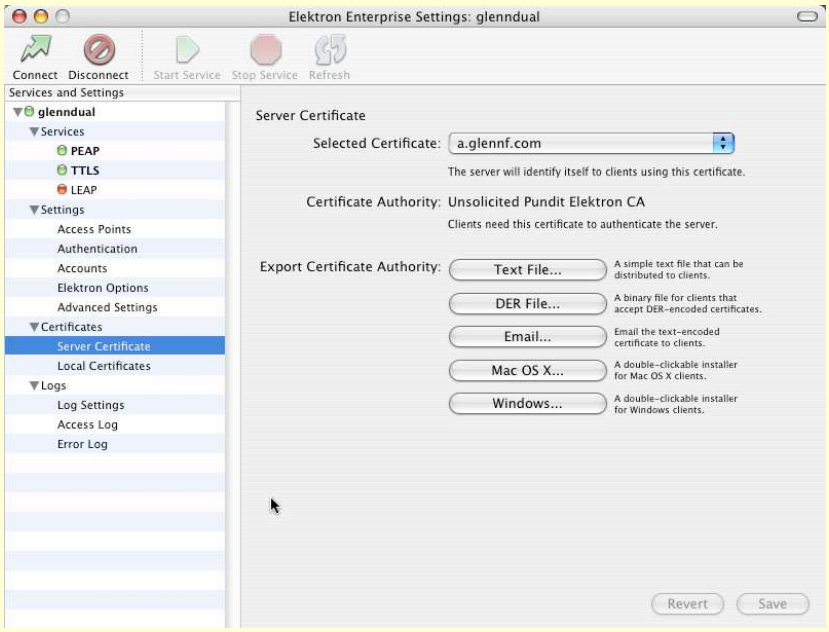


Enter a password that is used later as the *shared secret* in the access point's configuration.

4. Click the Identity tab. Here you export the certificate authority needed, in order to use the self-signed certificate created earlier on Wi-Fi client computers that will connect using this Elektron Server

(**Figure 22**). Install a certificate on each machine that will connect: it's easiest to create Mac OS X and Windows installers, copy them to those machines, and then run the installers on those machines.

FIGURE 22



The default certificate for this machine was installed when Elektron Setup Assistant created it.

Elektron Server is now fully configured for the following steps, although you can explore a number of additional settings for your particular installation.

Configure Your Base Station to Hand Off Credentials

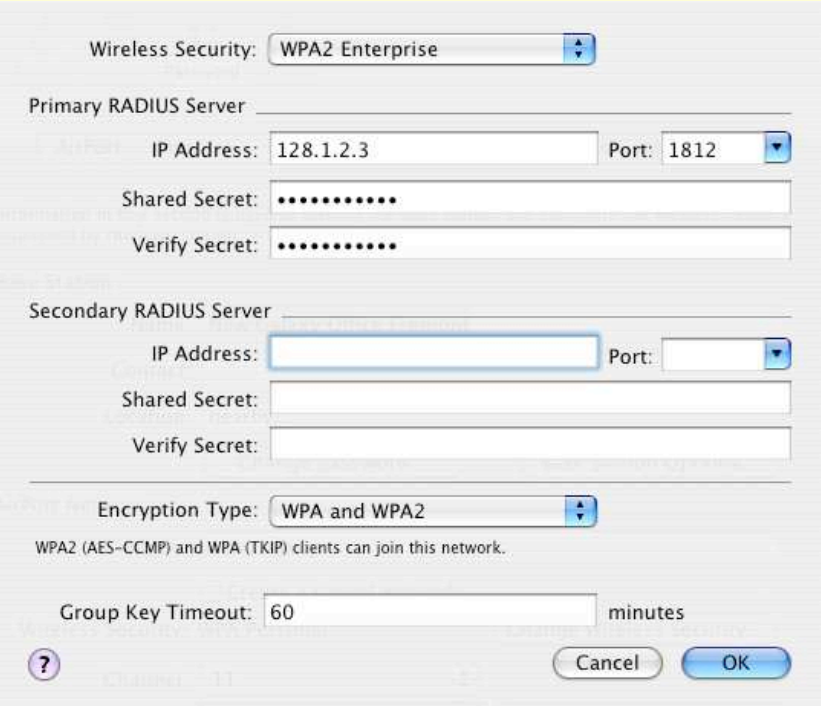
Your base station has to be set up to talk to the Elektron Server so that when a user wants to connect via WPA Enterprise, your base station can hand off their credentials—login name and password–to the server, and then recognize from the server's reply that the credentials are good or bad. These steps will hook your base station and an Elektron Server together:

1. Launch AirPort Admin Utility and connect to your base station.
2. In the AirPort pane, click the Change Wireless Security button.
3. Choose WPA Enterprise from the Wireless Security pop-up menu and click OK.

4. Enter the IP address, choose 1812 from the Port pop-up menu, and in the Shared Secret field, enter the access point password from Step 3 in the previous set of steps (**Figure 23**). Click OK.

TIP The most robust networks have two servers running with identical information; the secondary server is queried when the first is unavailable. Elektron Enterprise Edition offers this option at a much higher cost: $749 (for each server) instead of $299 each for the regular server.

FIGURE 23



Enter the Elektron Server's settings: its IP address and shared secret (the access point password). Set the port to 1812.

5. Click Update to change settings and reboot the base station.

Making these changes will render your AirPort network unavailable to any user that hasn't been given Elektron credentials, so be careful if you're just testing this mode.

Further, if you configured one element in this setup incorrectly, like entering the wrong IP address or password in the Access Control pane, you will have to connect to the base station via Ethernet to correct the error.

Connect with Panther or Tiger

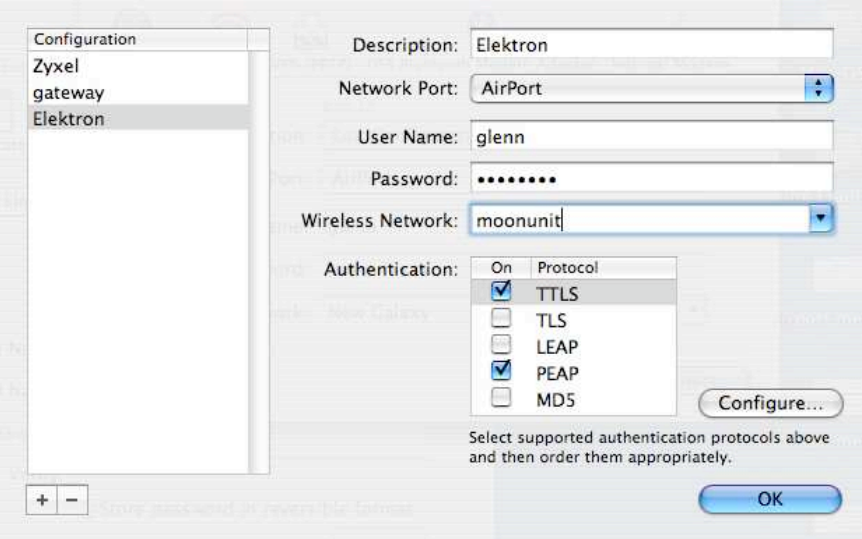
Now that you've configured Elektron Server, here's how to connect to it from Mac OS X 10.3 Panther or 10.4 Tiger.

1. In Internet Connect (find it in the **`/Applications`** folder), choose File > New 802.1X Connection.

3. From the Configuration pop-up menu, choose Edit Configurations.

4. Click the + (plus) button at the lower left and enter your user account details (**Figure 24**). In this case, choose AirPort as the network port, enter a valid Elektron user name and the account's associated password (not the access-point password), and choose the Elektron-connected network from the Wireless Network pop-up menu. Under Authentication, uncheck options that won't be used: Elektron supports both PEAP and EAP-TTLS, so you can leave both or either checked. Click OK when you finish.

FIGURE 24

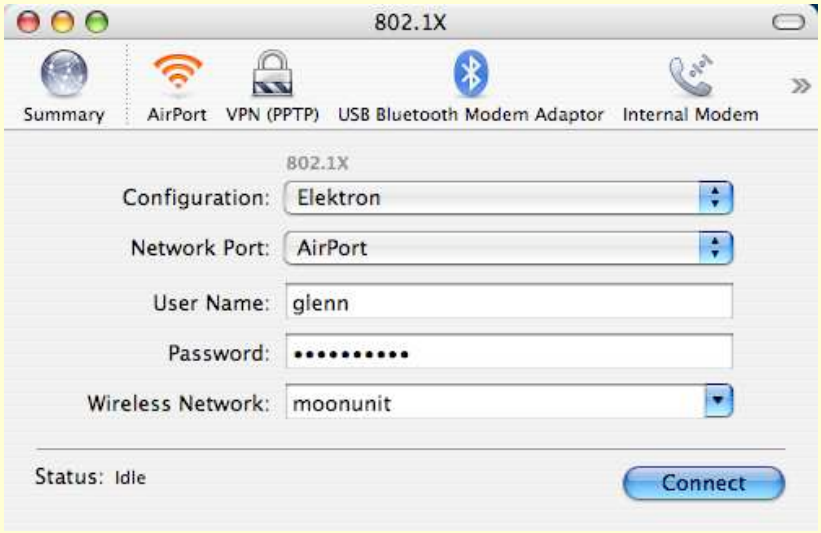


Setting up the basic configuration options for an 802.1X connection.

5. In the main 802.1X connection screen, click Connect to start a session (**Figure 25**).

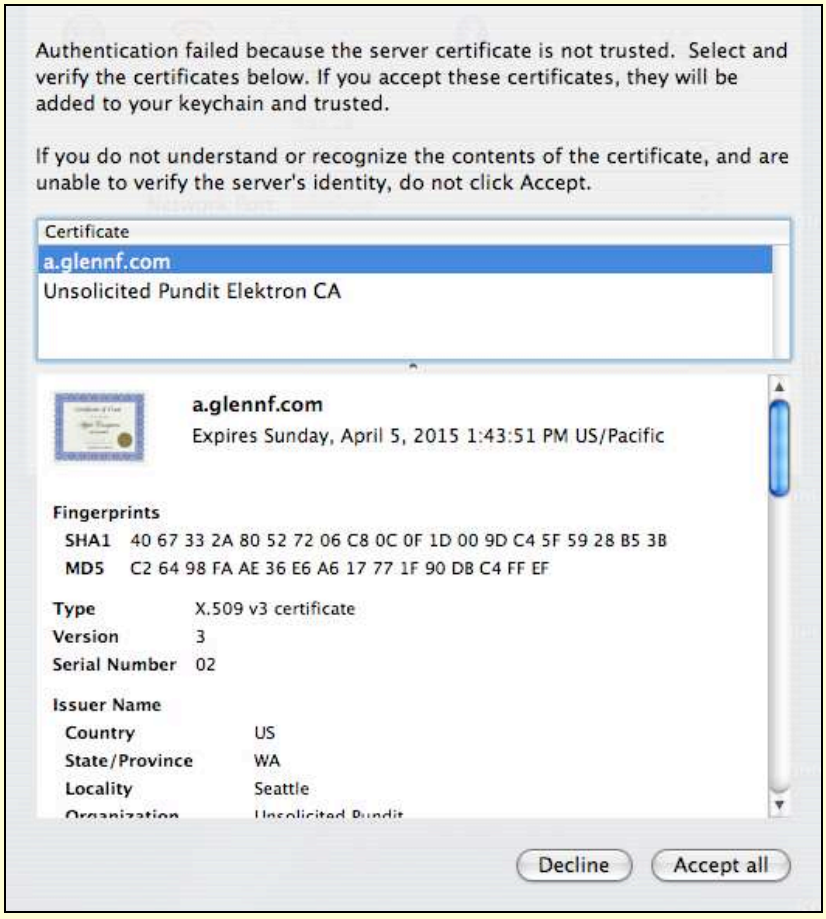
If you decided to forgo using the package installer in the previous set-up instructions to install a certificate authority, you can still connect to the network. The access point sends the two certificates—for the certificate authority and the specific Elektron Server—to Internet Connect, which presents you with details about them, and asks you to confirm that they're okay (**Figure 26**). Click Accept All to proceed.

FIGURE 25



Click Connect to start your session with an 802.1X-enabled Wi-Fi network.

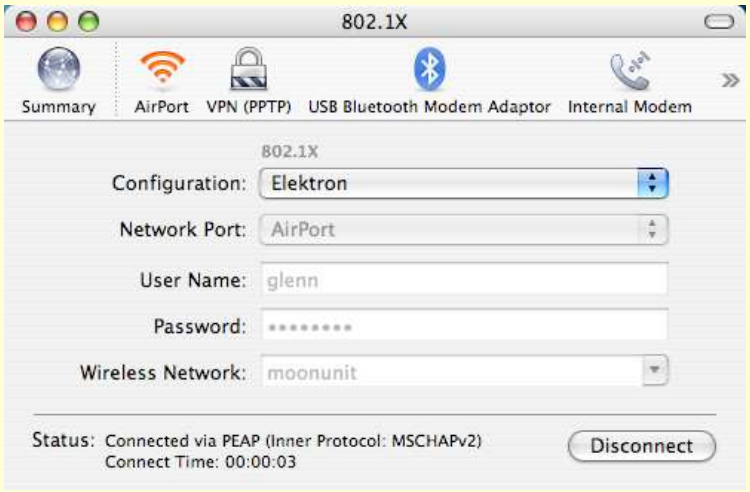
FIGURE 26



Confirm the certificates by clicking Accept All.

Finally, you're connected. In this case, Elektron uses Protected EAP (PEAP), and that protocol name is noted in the Status line of the 802.1X connection dialog (**Figure 27**).

FIGURE 27



When you're connected to a server using 802.1X, you can see the duration of the connection and the method by which the connection was made (in this case, PEAP). Click Disconnect to end the session.

The next time you connect, you won't be prompted for a certificate because it's already in your keychain. You'll see a prompt for that access point only if someone attempts to spoof it and convince you to connect to a masquerading, rogue access point. It won't happen in your home, but it occurs in businesses regularly. As public hotspots adopt 802.1X—T-Mobile did in 2004 and iBahn, a hotel Wi-Fi operator, added in 2005—this masquerade might have serious consequences.

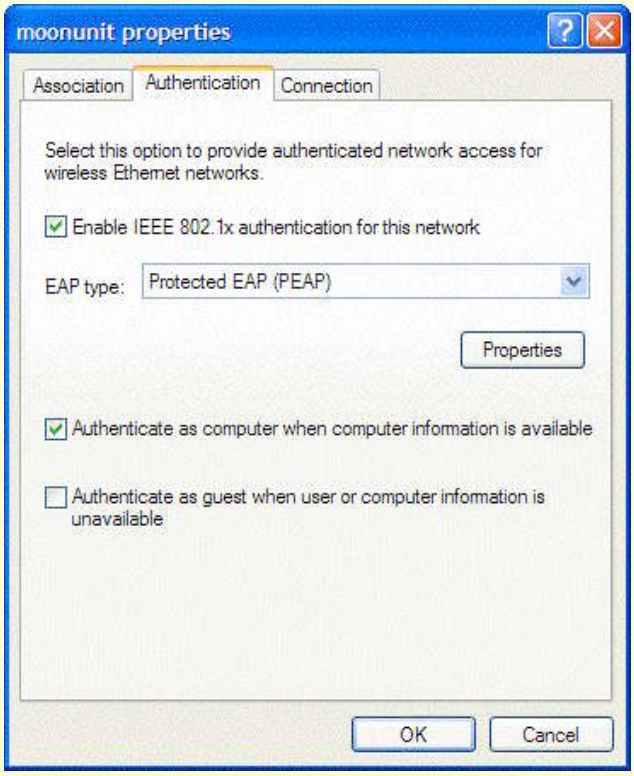
Connect with Windows XP Service Pack 2

It's easy to connect from Windows XP SP2 to a WPA Enterprise network. Follow these steps to create a secure connection:

1. If you haven't already, use the Identity pane of Elektron Settings to create an installer for a Windows-based digital certificate. (This is optional for a Mac, but mandatory under Windows.) Install the resulting certificate.
2. Right-click the wireless icon in the System Tray and choose Open Network Connections.
3. Select Wireless Network Connection, and then, at the left, click Change Settings of This Connection.
4. Click Add to create a new preferred network.
5. In the Association pane, enter the network's name, choose WPA from the Network Authentication pop-up menu, and leave Data Encryption set to TKIP unless you know that the network supports AES.

6. Click the Association tab, and choose Protected EAP (PEAP) from the EAP Type pop-up menu (**Figure 28**).

FIGURE 28



Make sure PEAP is selected from the EAP Type pop-up menu.

7. Click the Properties button under the EAP Type pop-up menu. Next to the Select Authentication Method pop-up menu, click Configure.
8. Uncheck Automatically Use My Windows Logon Name, unless your Windows user name and password are identical to the one you set up within Elektron Server.
9. Click OK, OK, OK to dismiss the nested dialogs—it's the Joe Pesci sequence.
10. Move the new network into the topmost position in Preferred Networks to have your computer connect to the network. You may need to bring up View Wireless Networks to select and connect to the network, however.
11. Enter your user name and password when prompted.

Appendix B: Password Advice

We talk blithely about passwords throughout this ebook, and more generally, passwords are all around us. But are you picking good passwords? A bad password can be cracked easily, often just by guesswork. So here's some advice.

Generate Three Passwords

With the understanding that it's nearly impossible to remember different passwords for every possible service, we recommend using three different passwords. If you restrict yourself to three passwords and always use the same email address or user name, the likelihood of forgetting your access information for any given site or program is low.

- **Low-Security:** Create a standard low-security password that's simple and easily remembered. Since it's low-security, make sure to use it only for Web sites that don't store personal information about you (such as your address, birth date, or credit card number). In essence, this password protects only your online identity; if someone were to guess it, they could pretend to be you in a discussion forum or the like.

- **Medium-Security:** For Web sites and accounts where some personal information is at risk, create a medium-security password. It will be harder to type, since it should include upper- and lowercase letters, numbers, and punctuation.

- **High-Security:** Everyone should have one highly secure password that is long, hard to type, and essentially impossible to guess. Use it for accounts, like your bank and PayPal, where money is involved, and for programs that store other passwords. Using a longer password won't prevent it from being stolen via an unprotected wireless transaction, but realistically, most passwords are stolen by being guessed or because someone wrote them on a Post-it note. You're also unlikely to need this password on a site that wouldn't secure the transaction, and in fact, don't use this password on sites that don't secure transactions.

TIP This approach may fall down in a situation where a family must share passwords needed to access joint financial information such as bank accounts, insurance policies, and so on. Unfortunately, many financial institutions restrict user name or password lengths in such a way that prevent you from using the system we've outlined.

In such a case, you must record user names and passwords in a place where everyone who needs to can access them. Some people may choose to use a notebook, but it's unfortunately easily stolen or lost. A more secure approach would be to use a program like Alco Blom's Web Confidential (available for Mac OS 8, Mac OS 9, Mac OS X, Windows, and Palm OS at http://www.web-confidential.com/). You can enter all your user names and passwords into Web Confidential, where they're encrypted and protected with yet another password, which you'd make sure was known to everyone who needed to know.

Be sure to discuss this situation with people in your family, since you never know when it could be important to access someone else's secure accounts (legitimately, of course) in case of severe illness or injury, or in the case of an elderly relative, mental incapacitation.

Learn to Create a Highly Secure Password

Many security experts now recommend that when you enter a really long password–often called a *passphrase* because it's composed of separate words or items—that you think of something memorable to you that no one else would know: a lyric of a song, for instance. Instead of choosing "754!#%kdja" you might enter "shall I compare thee 2 a summer's day?!"

The length makes it harder to crack while still rendering it memorable to you. The extra punctuation at the end (or wherever you choose to put it—even extra spaces between words would help) helps stymie efforts to crack the passphrase against a database of all poems and song lyrics.

While many systems require a short password, others like WPA and PGP allow dozens or even hundreds of characters. Systems with longer passphrases typically only need you to type them once per computer (to set up a secure Wi-Fi network) or once per session (each time you reboot, for instance).

SIDEBAR MAC OS X KEYCHAIN

If you store passwords in the Keychain in Mac OS X, note that by default the Keychain uses the same password as your login, which might not be one of your more secure passwords since you have to enter it so frequently. But, using the Keychain Access program (found in **`/Applications/Utilities`**) you can change the password for your Keychain to something more secure; just choose Edit > Change Password for Keychain "*username*" and enter a new password.

The version of Keychain Access in Tiger (Mac OS X 10.4) also has a tool that can help you figure out how secure the password you choose is. Choose File > New Password Item, and then click the key icon to the right of the Password field in the New Password Item dialog. The Password Assistant window that appears suggests passwords of varying strengths or tests ones you enter.

In theory, you could generate highly secure random passwords with Keychain Access every time you needed a password for a Web site. That way, if a password for one site is made public, it won't compromise any other sites. The downsides are that it's more of a fuss and you're putting all your eggs in the Keychain's basket, requiring an extremely solid backup strategy.

GLOSSARY

802.11b: The most common of the three wireless networking specifications included in the Wi-Fi certification mark. 802.11b uses the 2.4 GHz band and runs at up to 11 Mbps.

802.11g: The newest of the three Wi-Fi specifications. 802.11g is backward compatible with 802.11b, thanks in part to its use of the 2.4 GHz band, and it runs at up to 54 Mbps. Most new equipment uses 802.11g.

802.11i: A security standard designed to replace the broken WEP (Wired Equivalent Privacy) integrity and encryption system that was part of the original 802.11b specification. 802.11i is implemented in part by WPA (Wi-Fi Protected Access) and in full by the later WPA2 standard.

802.1X: An authentication specification that allows a client to connect to a wireless access point or wired switch but prevents the client from gaining access to the Internet until it provides credentials, like a user name and password, that are verified by a separate server. In 802.1X, there are three roles: the supplicant (client), authenticator (switch or access point), and authentication server.

access point: The hub of a wireless network. Wireless clients connect to the access point, and traffic between two clients must travel through the access point. Access points are often abbreviated to AP in industry literature, and you may also see them referred to as "wireless routers," "wireless gateways," and "base stations."

AES-CCMP: An extremely strong encryption standard that's just starting to become available. AES stands for Advanced Encryption System. CCMP is a complex abbreviation: Counter-mode CBC-MAC Protocol; CBC-MAC stands for Cipher Block Chaining-Message Authentication Code. You don't need to know this unless you attend highly geeky cocktail parties.

AirPort Extreme: Apple's marketing name for its 802.11g wireless networking technology (which includes the AirPort Express Base Station).

AirPort: Apple's original marketing name for its 802.11b wireless networking technology. Today, AirPort refers to Wi-Fi-compatible wireless networking in general for Apple.

APOP: A protocol for protecting email passwords used with POP. APOP stands for Authenticated Post Office Protocol.

authenticate: The process of confirming the identity of someone connecting to a network.

authentication server: A back-end database server that confirms the identity of a supplicant to an authenticator in an 802.1X-authenticated network.

authenticator: The gatekeeper role in an 802.1X-authenticated network. You can think of the authenticator as a gatekeeper; access points and Ethernet switches can act as authenticators.

base station: See *wireless gateway*.

certificate authority: A trusted third party that can assure the identity of others when using security systems like SSL. A certificate authority registers the digital identity of a site or individual, and lets you confirm manually or automatically that someone you're interacting with—say, over a secure Web connection—is who he appears to be.

certificate: A computer-readable credential. Certificates are typically signed by other people or certificate authorities to guarantee their authenticity.

clear text: Sensitive information like passwords sent across a network without encryption. Clear text is also commonly referred to as "in the clear."

cloning: The act of replicating one device's MAC address onto another to work around restrictions that prevent only particular MAC addresses from connecting to a network. Also sometimes called "spoofing."

closed network: A wireless network that doesn't advertise its network name.

DMZ: A feature in a NAT gateway that lets you expose a machine on your internal network to the outside Internet. DMZ nominally stands

for *demilitarized zone,* and is sometimes also called "virtual server." It's basically port mapping for all available ports.

EAP: A standard form of generic messaging used in 802.1X, among other places. EAP stands for Extensible Authentication Protocol.

EAP-TLS: Used to create a secured connection for 802.1X by pre-installing a digital certificate on the client computer. EAP-TLS stands for Extensible Authentication Protocol-Transport Layer Security.

ESSID: Extended Service Set Identifier. See *network name*.

fingerprint: A short sequence of characters you can send someone so she can verify that a specific public key is actually your public key.

firewall: A network system that blocks malevolent or unauthorized traffic that might endanger the computers on your network.

firmware: The internal software that runs dedicated hardware devices. Upgrades to firmware are often necessary to fix problems.

gateway: See *wireless gateway*.

hot spot: A place where you can connect to a public wireless network.

HTTP: The network protocol used by the Web, although it's also now used for many other services. HTTP stands for Hypertext Transfer Protocol.

HTTPS: The SSL-protected version of HTTP.

IMAP: An increasingly common way of receiving email from a mail server on the Internet. IMAP defaults to storing mail on a server, in contrast to POP, which stores mail on your computer. IMAP stands for Internet Message Access Protocol.

IPsec: One of the main protocols used for VPNs. IPsec stands for IP security.

key server: An Internet-based server that lets you look up other people's public keys.

local area network: The computers at your site, connected via Ethernet or Wi-Fi. Local area network is often abbreviated to LAN. Compare local area networks with wide area networks.

MAC address: The unique address assigned to every wireless and wired Ethernet network adapter. MAC stands for Media Access Control. Despite the fact that assigned MAC addresses are all unique, it's possible to assign one device's MAC address to another device. There are various reasons (to circumvent ISP restrictions) to clone MAC addresses.

NAT: A network service that makes it possible to share a single IP address with a network of many computers. NAT stands for Network Address Translation. Since a NAT gateway exposes only a single IP address to the outside Internet, it's useful for security, and some manufacturers may call it, somewhat incorrectly, a "firewall."

network adapter: The card or built-in hardware used in a computer or handheld device to connect to a network, whether wired or wireless.

network name: The name you give network; it's what shows up when a wireless client displays available networks. Many manufacturers use the terms "SSID" or "ESSID" in place of network name.

open network: A wireless network that is broadcasting its name. Technically, the fact that a network is broadcasting its name is unrelated to whether or not it employs WEP or WPA encryption, but informally, an open network is often considered one that can be used by anyone, without a password.

pass-through: See *port mapping*.

personal certificate: A certificate you generate for use with SSL that doesn't have a certificate authority behind it. Personal certificates, also known as "self-signed certificates," aren't vouched for by a certificate authority, but they're good enough in cases where you're working with private SSL-enabled systems.

PEAP: A method of securing an 802.1X session within an encrypted tunnel to protect credentials used for logging in. PEAP stands for Protected Extensible Authentication Protocol.

PGP: A technology and set of programs for encrypting data. PGP stands for Pretty Good Privacy.

plain text: See *clear text*.

POP: The most common way of receiving email from a mail server on the Internet. POP defaults to storing mail on your computer, in contrast to IMAP, which stores mail on the server. POP stands for Post Office Protocol.

port forwarding: See *port mapping*.

port mapping: The act of mapping a port on an Internet-accessible NAT gateway to another port on a machine on your internal network. Port mapping enables you to run a public Internet service on a machine that is otherwise hidden from the Internet by your NAT gateway. Other names for port mapping include "port forwarding," "pass-through," and "punch-through."

port: Either a physical jack on a network device or a way of identifying the type of data being sent in an Internet connection. Every Internet service has its own port number.

PPTP: A Microsoft-developed protocol used for VPNs that is easily used from within Windows and Mac OS X. PPTP stands for Point-to-Point Tunneling Protocol.

pre-shared key: A *TKIP* passphrase used to protect your network traffic in WPA. Some manufacturers use the term "pre-shared secret" instead.

pre-shared secret: See *pre-shared key*.

private key: The key you keep secret in public-key cryptography systems. You use your private key to decrypt encrypted data sent to you by other people, who used your public key to encrypt it. You also use your private key to sign email messages; your recipients then use your public key to verify your signature.

promiscuous mode: A state of a wireless network adapter in which it listens to all the traffic on a wireless network rather than just the traffic addressed to your computer.

public key: The key you give out to the world in public-key cryptography systems. Other people use your public key when sending you encrypted data, which you can then decrypt with your private key. You also use other people's public keys to verify the authenticity of mail messages they've signed with their private keys.

relaying: The act of sending email through your mail server when you're not connected to your local network. Spammers take advantage of mail servers that allow unrestricted relaying.

script kiddies: Wanna-be crackers who don't have the technical skills to break into computers on their own, so they use canned cracking software.

self-signed certificate: See *personal certificate*.

SMTP AUTH: A command in the SMTP protocol that provides identification to an SMTP server, so it will accept outgoing mail from you. SMTP AUTH is essentially authenticated SMTP.

SMTP: The protocol for sending email on the Internet. SMTP stands for Simple Mail Transfer Protocol.

SSH: A security system that lets you create encrypted tunnels for any Internet protocol via port forwarding. SSH stands for Secure Shell.

SSID: Service Set Identifier. See *network name*.

SSL: A security protocol that secures Internet transactions at the program level. SSL, which stands for Secure Sockets Layer, is widely used in Web browsers to protect credit card transactions, for instance. SSL is a component in EAP-TLS (Extensible Authentication Protocol-Transport Layer Security). SSL is also increasingly used for VPNs. SSL is the predecessor to TLS, but because the two standards are very similar and because TLS is backward compatible to the last version of SSL, the two terms are used interchangeably.

stumbler: A software program that looks for available wireless networks in range and reports information about them.

supplicant: The client role in an 802.1X-authenticated network.

TKIP: An encryption key that's part of WPA. TKIP stands for Temporal Key Integrity Protocol. It's nominally weaker than the government-grade AES, but in the real world, TKIP is more than strong enough.

TLS: Transport Layer Security. See *SSL*.

trigger: A special form of port mapping in which outgoing traffic on a specific port alerts a NAT gateway to allow incoming traffic on other ports. Triggers are used for network gaming.

VPN: A method of creating an encrypted tunnel through which all traffic passes, preventing anyone from snooping through transmitted and received data. VPN stands for virtual private network.

WEP: An encryption system for preventing eavesdropping on wireless network traffic. WEP stands for Wired Equivalent Privacy. WEP is easily broken, and is in the process of being replaced by WPA.

Wi-Fi: A certification mark managed by a trade group called the Wi-Fi Alliance. Wi-Fi certification encompasses numerous standards, including 802.11a, 802.11b, 802.11g, WPA, and more, and equipment must pass compatibility testing to receive the Wi-Fi mark.

Wi-Fi Protected Access: See *WPA*.

wireless gateway: A generic term that we use to differentiate between an access point and a more-capable device that can share an Internet connection, serve DHCP, and bridge between wired and wireless networks. You may also see the term "wireless router," or "base station."

wireless network adapter: See *network adapter*.

WPA: A modern encryption system for preventing eavesdropping on wireless network traffic that solves the problems that plague WEP. WPA stands for Wi-Fi Protected Access. WPA is a subset of the IEEE *802.11i* security standard, but WPA was released before that standard was finalized. It includes the *TKIP* encryption key.

WPA2: WPA2 is a superset of *WPA* that includes additional security measures; it's a complete implementation of the IEEE *802.11i* security standard. WPA2 includes support for *TKIP* and *AES-CCMP*.

zombie: A computer that has been taken over by a malevolent program that uses it to attack other computers.

About This Ebook

In contrast to traditional print books, Take Control ebooks offer clickable links, full-text searching, and free minor updates. We hope you find them useful and enjoyable to read. Keep reading in this section to learn more about the authors, the Take Control series, and the publisher.

About Glenn

Photograph Copyright ©1997 Karen Moskowitz. Used with permission.

Glenn Fleishman has written for hire since 1994, starting with *Aldus Magazine*. He contributes regularly to *Macworld*, *Mobile Pipeline*, *Popular Science*, *The Economist, The New York Times*, and *The Seattle Times*. He's one of the two Macintosh columnists for *The Seattle Times*, and a contributing editor at *TidBITS*.

Glenn spends much of his time writing about wireless networking. He co-wrote two editions of *The Wireless Networking Starter Kit* with Adam Engst (Peachpit Press, 2002 and 2003). He edits the daily Web log Wi-Fi Networking News (http://www.wifinetnews.com/) and five related wireless blogs, and he is the senior editor of Jiwire (http://www.jiwire.com/).

About Adam

Adam C. Engst is the publisher of *TidBITS,* one of the oldest and most respected Internet-based newsletters, distributed weekly to many thousands of readers. He has written numerous technical books, including the best-selling *Internet Starter Kit* series, and many magazine articles (thanks to Contributing Editor positions at *MacUser, MacWEEK,* and now *Macworld*).

Adam's innovations include the creation of the first advertising program to support an Internet publication (in 1992), the first flat-rate accounts for graphical Internet access (in 1993, with Northwest Nexus for *Internet Starter Kit for Macintosh*), and the new Take Control electronic book series. In addition, he has collaborated on several Internet educational

videos and has appeared on a variety of internationally broadcast television and radio programs.

Adam's indefatigable support of the Macintosh community and commitment to helping individuals has resulted in numerous awards and recognition at the highest levels. In the annual MDJ Power 25 survey of industry insiders from 2000 through 2004, he ranked in the top five most influential people in the Macintosh industry, and he was named one of *MacDirectory's* top ten visionaries. And how many industry figures can boast of being turned into an action figure?

Authors' Acknowledgements

Thanks to Tonya for all she does, both in editing this title and in keeping Take Control running.

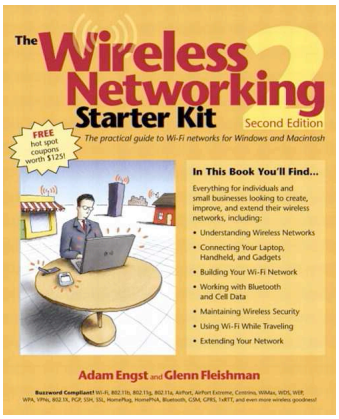
A tip of the mouse to Chris Pepper, Larry Rosenstein, and Joe Kissell for their excellent comments during our collaborative editing phase.

Shameless Plug

If you liked this title, you'll undoubtedly like our other works:

- ***TidBITS***
  For award-winning Macintosh commentary and editorial from both Adam and Glenn, be sure to subscribe (for free!) to *TidBITS*.
  http://www.tidbits.com/about/list.html

- ***The Wireless Networking Starter Kit***
  For more information about wireless networking in general, refer to our definitive book, which covers everything for individuals and small businesses looking to create, improve, and extend their wireless networks, including details about Wi-Fi networks, Bluetooth, cellular data, wireless security, long-range wireless networking, using Wi-Fi on the road, and much more.
  http://wireless-starter-kit.com/



Take Control: The Series

Take control of computing with the Take Control series of highly practical, tightly focused electronic books! Written by leading Macintosh authors, edited by TidBITS, and delivered to your electronic doorstep within moments of "going to press," the Take Control ebooks provide just the technical help you need. http://www.takecontrolbooks.com/

- *Take Control of Your AirPort Network,* by Glenn Fleishman http://www.takecontrolbooks.com/AirPort.html
- *Take Control of Sharing Files in Tiger,* by Glenn Fleishman http://www.takecontrolbooks.com/tiger-sharing.html
- *Take Control of Mac OS X Backups,* by Joe Kissell http://www.takecontrolbooks.com/backup-macosx.html
- *Take Control of Buying a Mac,* by Adam C. Engst http://www.takecontrolbooks.com/buying-mac.html

About TidBITS Electronic Publishing

Take Control ebooks are a project of TidBITS Electronic Publishing. TidBITS Electronic Publishing has been publishing online since 1990 when publishers Adam and Tonya Engst first created their online newsletter, *TidBITS,* about Macintosh and Internet-related topics. *TidBITS* has been in continuous, weekly production since then (http://www.tidbits.com/).

Adam and Tonya are well known in the Macintosh world as writers, editors, and speakers, and they have written innumerable online and print publications. They are also parents to Tristan, who thinks ebooks about trains, ships, and dinosaurs would be cool.

Production Credits

- Cover: Jeff Carlson, http://www.necoffee.com/
- Take Control logo: Jeff Tolbert, http://jefftolbert.com/
- Editor in Chief: Tonya Engst, http://www.tidbits.com/tonya/
- Publisher: Adam Engst, http://www.tidbits.com/adam/

Take Control of Your Wi-Fi Security

ISBN: 0-9759503-9-8

TidBITS Electronic Publishing
50 Hickory Road
Ithaca, NY 14850 USA
http://www.takecontrolbooks.com/

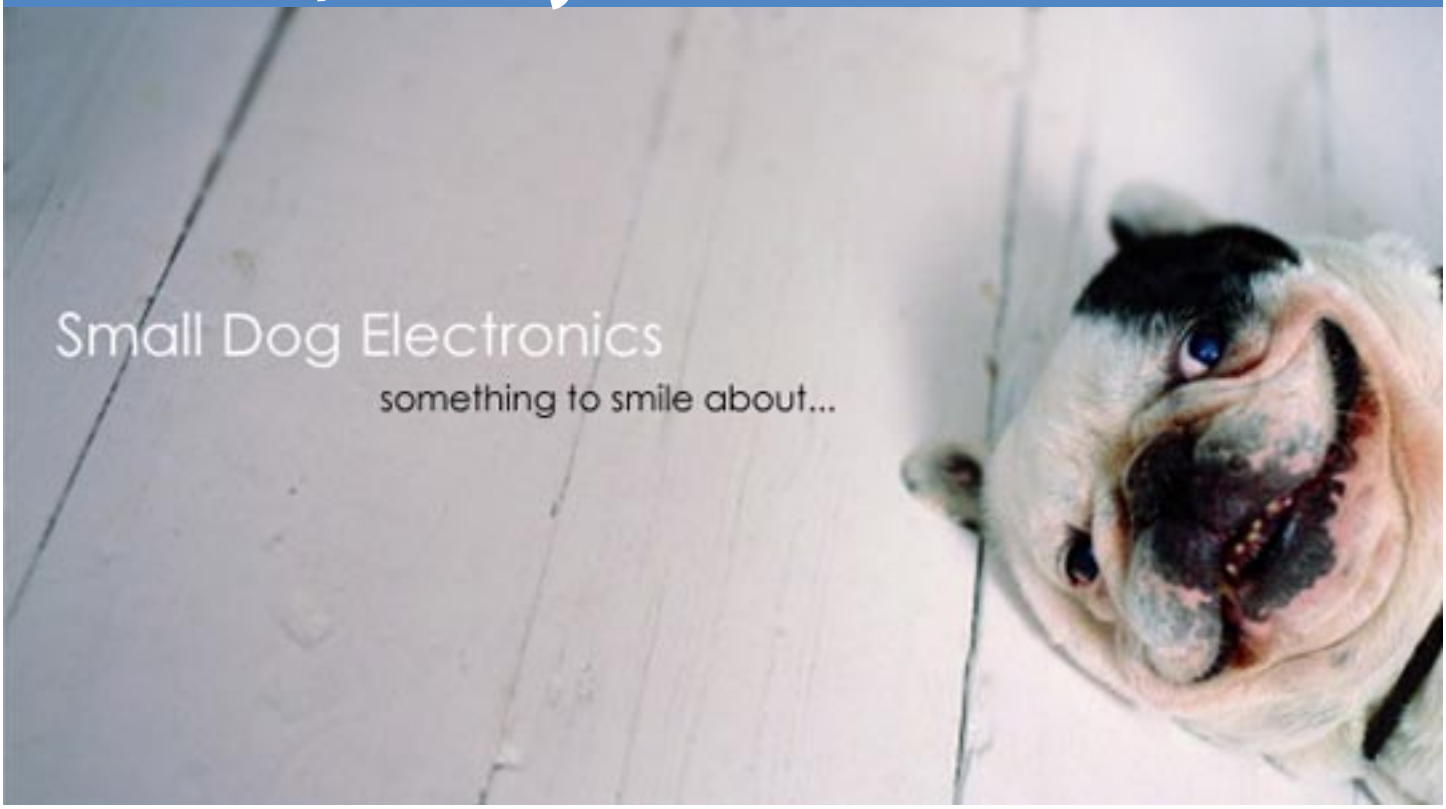
September 2005. Version 1.0

Take Control ebooks help readers regain some measure of control in an oftentimes out-of-control universe. Take Control ebooks also streamline the publication process so that information about quickly changing technical topics can be published while it's still relevant and accurate. Send comments about this, or any, Take Control ebook to tc-comments@tidbits.com.

What do you get when you buy from Small Dog Electronics?

service selection value expert advice small dogs!

Now, $5 off your next web order!



Small Dog Electronics
www.smalldog.com

Redeem your coupon on-line at www.smalldog.com Limited to one use per customer. Enter coupon # bone92624125 at check out.

Apple Specialist

1-800-511-MACS
We measure success in more than just dollars.