CHAPTER 4

HACKING WINDOWS 95/98 AND ME

The most important thing for a network administrator or end user to realize about Windows 95/95B/98/98SE and their updated counterpart Windows Millennium Edition (hereafter Win9*x*/Me, or the "DOS Family") is that their architecture was not designed to incorporate security from the ground up like Microsoft's other Windows lineage, the Windows NT Family.

NOTE Throughout this book, we use the phrase "NT Family" to refer to all systems based on Microsoft's New Technology (NT) platform, including Win NT 3.*x*–4.*x*, Windows 2000, Windows XP, and Windows .NET Server (see Chapter 5). Where necessary, we will differentiate between desktop and server versions. In contrast, we will refer to the Microsoft DOS/Windows 1.*x*/3.*x*/9*x*/Me lineage as the "DOS Family."

In fact, it seems that Microsoft went out of its way in many instances to sacrifice security for ease of use when planning the architecture of Win9*x*/Me. This becomes double jeopardy for administrators and security-unaware end users. Not only is Win9*x*/Me easy to configure, but the people most likely to be configuring it are also unlikely to take proper precautions (such as good password selection).

Even worse, unwary users of Win9*x*/Me could be providing a back door into your corporate LAN, or they could be storing sensitive information on a home PC connected to the Internet. The growing prevalence of viruses and other Web- or e-mail-borne malicious software that "phone home" from compromised systems complicates this issue. A single unsuspecting Windows 9*x* user who launches a malicious e-mail attachment can create a tunnel back out of the firewall to a malicious network, setting the stage for a full-scale invasion.

With the increasing adoption of cable and DSL high-speed, always-on Internet connectivity, this problem will only get worse. Whether you are an administrator who manages Windows 9*x* or a user who relies on Windows 9*x* to navigate the Net and access your company's network from home, you need to understand the tools and techniques that will likely be deployed against you.

Fortunately, Win9*x*/Me's simplicity also works to its advantage security-wise. Because it was not designed to be a true multiuser operating system, it has extremely limited remote administration features. It is impossible to execute commands remotely on Win9*x*/Me systems using built-in tools, and remote access to the Windows 9*x* Registry is only possible if access requests are first passed through a security provider such as a Windows NT Family server or Novell NetWare server. The NT Family and Novell NetWare provide *user-level* security, versus the locally stored, username/password-based *share-level* security that is the default behavior of Win9*x*/Me. (Win9*x*/Me cannot act as a user-level authentication server.)

Therefore, Win9*x*/Me security is typically compromised via the classic routes: misconfiguration, tricking the user into executing code, and gaining physical access to the console. We have therefore divided our discussions in this chapter along these lines: remote and local attacks. We also cover Windows 9*x* separately from Windows Me because the two OSes were released over three years apart. However, in most instances, attacks against Windows 9*x* should also work against Windows Me, unless otherwise specified.

If you are a Win9*x*/Me user wondering whether you should upgrade to Microsoft's newest desktop operating system, Windows XP, we'll say, in a word, YES! XP has all the

Plug-and-Play warmth that novice users covet with ten times the stability and an actual security subsystem, because it is based on the NT Family code lineage, as opposed to the DOS Family. Either the Home Edition or Professional is appropriate, depending on whether you want a more simplified default user interface with plenty of helpful wizards or need more business-oriented features, such as Remote Desktop, System Restore, and advanced networking features. We discuss Windows XP and its business-oriented cousins, Windows NT, Windows 2000, and .NET Server 2003 in Chapter 5.

NOTE Win9x/Me is rightfully classified as an end-user platform. Often, the easiest way to attack such a system is via malicious web content or e-mails directed at the user rather than the operating system. Therefore, we highly recommend reading Chapter 16, "Hacking the Internet User," in conjunction with this one.

WINDOWS 9*x* REMOTE EXPLOITS

Remote exploitation techniques for Windows 9*x* fall into four basic categories: direct connection to a shared resource (including dial-up resources), installation of backdoor server daemons, exploitation of known server application vulnerabilities, and denial of service. Note that three of these situations require some misconfiguration or poor judgment on the part of the Windows 9*x* system user or administrator and are therefore easily remedied.

Direct Connection to Windows 9*x* Shared Resources

This is the most obvious and easily breached doorway into a remote Windows 9*x* system. Windows 9*x* provides three mechanisms for direct access to the system: file and print sharing, the optional dial-up server, and remote Registry manipulation. Of these, remote Registry access requires fairly advanced customization and user-level security and is rarely encountered on systems outside of a corporate LAN.

One skew on the first mechanism of attack is to observe the credentials passed by a remote user connecting to a shared resource on a Windows 9*x* system. Because users frequently reuse such passwords, this often yields valid credentials on the remote box as well. Even worse, it exposes other systems on the network to attack.

Hacking Windows 9*x* File and Print Sharing

Popularity:	*8*
Simplicity:	*9*
Impact:	*8*
Risk Rating:	***8***

We've already covered some tools and techniques that intruders might use for scanning networks for Windows file and print shares (see Chapter 3) and noted that some of these also have the ability to attempt password-guessing attacks on the potential entry points. One of these tools is Legion from the Rhino9 group. Besides the ability to scan an

IP address range for Windows shares, Legion also comes with a brute force (BF) tool that will guess passwords provided in a text file and automatically map those that it correctly guesses. This is more correctly called a *dictionary attack* because it is based on a password list. One tip: the Save Text button in the main Legion scanning interface dumps found shares to a text file list, thus facilitating cutting and pasting into the BF tool's Path parameter text box, as Figure 4-1 shows.

The damage that intruders can do depends on the share that is now mounted. Critical files may exist in that directory, or some users may have shared out their entire root partition, making the life of the hackers easy indeed. They can simply plant devious executables into %systemroot%\Start Menu\Programs\Startup. At the next reboot, this code will be launched. (See upcoming sections in this chapter on Back Orifice for an example of what malicious hackers might put in this directory.) Alternatively, the .PWL file(s) can be obtained for cracking, as discussed later in this chapter.

File Share Hacking Countermeasures

Fixing this problem is easy—turn off file sharing on Windows 9*x* machines! For the system administrator who's worried about keeping tabs on a large number of systems, we suggest using the System Policy Editor (POLEDIT.EXE) utility to disable file and print sharing across all systems. POLEDIT.EXE, shown in Figure 4-2, is available with the Windows 9*x* Resource Kit (Win9*x* RK) but can also be found in the \tools\reskit\netadmin directory on most Win9*x* CD-ROMs or at http://support.microsoft.com/support/kb/articles/ Q135/3/15.asp.

If you must enable file sharing, use a complex password of eight alphanumeric characters (the maximum allowed by Windows 9*x*) and include metacharacters (such as [! @ # $ % &)

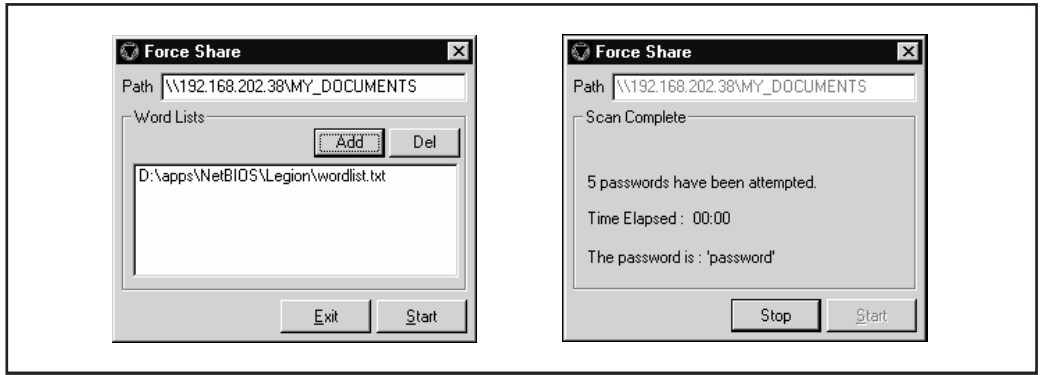


Figure 4-1. Legion's BF tool guesses Windows share passwords.

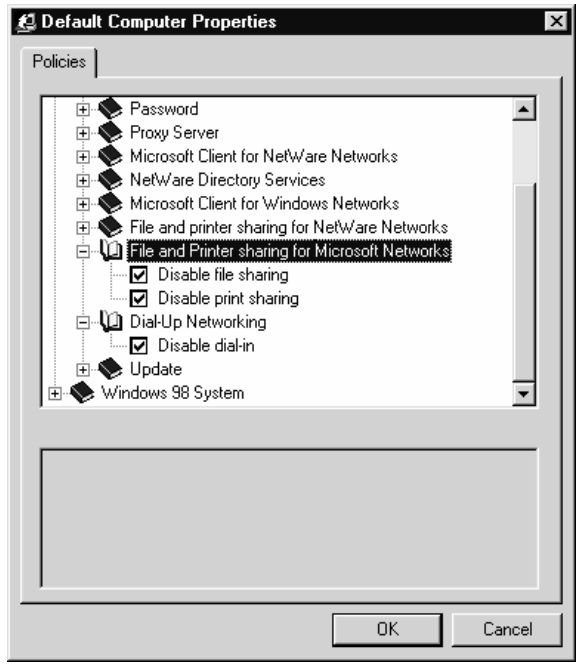


Figure 4-2. The Windows 9*x* System Policy Editor allows network administrators to prevent users from turning on file sharing or dial-in.

or nonprintable ASCII characters. It's also wise to append a dollar ($) symbol, as Figure 4-3 shows, to the name of the share to prevent it from appearing in Network Neighborhood, in the output of `net view` commands, and even in the results of a Legion scan.

Replaying the Windows 9*x* Authentication Hash

Popularity:	*8*
Simplicity:	*3*
Impact:	*9*
Risk Rating:	***7***

On January 5, 1999, the security research group known as the L0pht released a security advisory that pointed out a flaw in the Windows 9*x* network file sharing authentication routines (see http://www.atstake.com/research/advisories/1999/95replay.txt). While

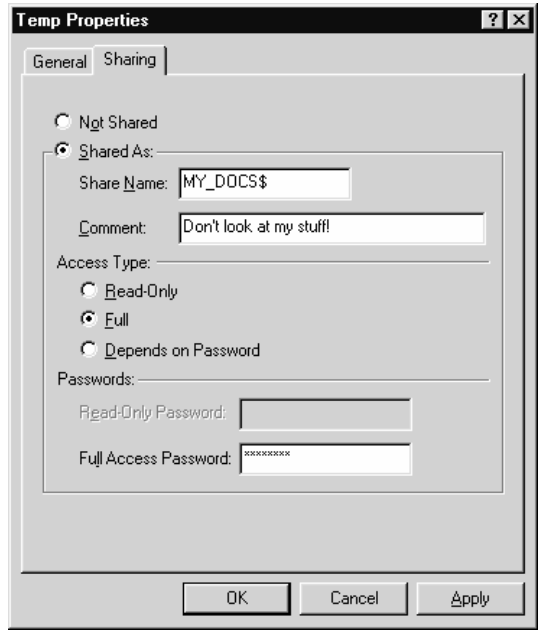


Figure 4-3. Append a $ symbol to the name of a file share to prevent it from appearing in Network Neighborhood and in the output of many NetBIOS scanning tools.

testing the new release of their notorious L0phtcrack password-eavesdropping and cracking tool (see Chapter 5), they noted that Windows 9*x* with file sharing enabled reissues the same "challenge" to remote connection requests during a given 15-minute period. Windows uses a combination of the username and this challenge to *hash* (cryptographically scramble) the password of the remote user, and the username is sent in cleartext. Attackers could simply re-send an identical hashed authentication request within the 15-minute interval and successfully mount the share on the Windows 9*x* system. During that period, the hashed password value will be identical.

Although this is a classic cryptographic mistake that Microsoft should have avoided, it is difficult to exploit. The L0pht advisory alludes to the possibility of modifying the popular Samba Windows networking client for UNIX (http://www.samba.org/) to manually reconstruct the necessary network authentication traffic. The programming skills inherent in this endeavor, plus the requirement for access to the local network segment to eavesdrop on the specific connection, probably set too high a barrier for widespread exploitation of this problem.

Hacking Windows 9*x* Dial-Up Servers

Popularity:	*8*
Simplicity:	*9*
Impact:	*8*
Risk Rating:	*8*

The Windows Dial-Up Server applet included with Windows 9*x*, shown in Figure 4-4, is another one of those mixed blessings for system admins. Any user can become a back door into the corporate LAN by attaching a modem and installing the inexpensive Microsoft Plus! for Windows 95 add-on package, which includes the Dial-Up Server components. (It now comes with the standard Windows 98 distribution.)

A system so configured is almost certain to have file sharing enabled, because this is the most common way to perform useful work on the system. It is possible to enumerate and guess passwords (if any) for the shares on the other end of the modem, just as we demonstrated over the network in the previous section on file share hacking, assuming that no dial-up password has been set.

Windows 9*x* Dial-Up Hacking Countermeasures

Not surprisingly, the same defenses hold true: Don't use the Windows 9*x* Dial-Up Server, and enforce this across multiple systems with the System Policy Editor. If dial-up capability is absolutely necessary, set a password for dial-in access, require that it be encrypted

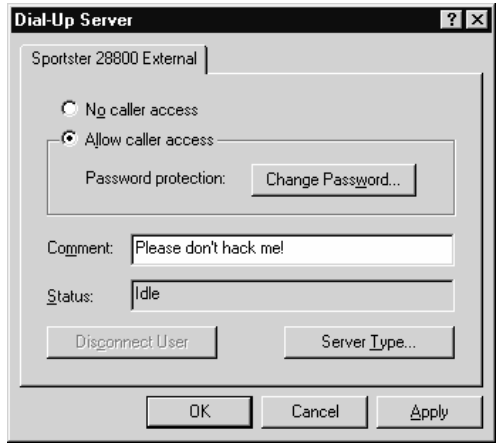


Figure 4-4. Making a Windows 9*x* system a dial-up server is as easy as 1-2-3.

using the Server Type dialog box in the Dial-Up Server Properties, or authenticate using user-level security. (That is, pass through authentication to a security provider such as a Windows NT domain controller or NetWare server.) Set further passwords on any shares (using good password complexity rules) and hide them by appending the $ symbol to the share names.

Intruders who successfully crack a Dial-Up Server and associated share passwords are free to pillage whatever they find. However, they will be unable to progress further into the network because Windows 9*x* cannot route network traffic.

It's also important to remember that Dial-Up Networking (DUN) isn't just for modems anymore. Microsoft bundles in Virtual Private Networking (VPN) capabilities (see Chapter 9) with DUN, so we thought we'd touch on one of the key security upgrades available for Win95's built-in VPN capabilities. It's called Dial-Up Networking Update 1.3 (DUN 1.3), and it allows Windows 95 to connect more securely with Windows NT VPN servers. This is a no-brainer: If you use Microsoft's VPN technology, get DUN 1.3 from http://support.microsoft.com/support/kb/articles/Q191/4/94.ASP. DUN 1.3 is also critical for protecting against denial of service (DoS) attacks, as you will see shortly.

We'll discuss other dial-up and VPN vulnerabilities in Chapter 8.

Remotely Hacking the Windows 9*x* Registry

Popularity:	*2*
Simplicity:	*3*
Impact:	*8*
Risk Rating:	***4***

Unlike Windows NT, Windows 9*x* does not provide the built-in capability for remote access to the Registry. However, remote access is possible if the Microsoft Remote Registry Service is installed (found in the \admin\nettools\remotreg directory on the Windows 9*x* distribution CD-ROM). The Remote Registry Service also requires user-level security to be enabled and therefore will at least require a valid username for access. If attackers were lucky enough to stumble upon a system with the Remote Registry installed, gain access to a writable shared directory, and were furthermore able to guess the proper credentials to access the Registry, they'd basically be able to do anything they want to the target system. Does this hole sound easy to seal? Heck, it sounds hard to create to us. If you're going to install the Remote Registry Service, pick a good password. Otherwise, don't install the service, and sleep tight knowing that remote Windows 9*x* Registry exploits just aren't going to happen in your shop.

Windows 9*x* and Network Management Tools

Popularity:	*3*
Simplicity:	*9*
Impact:	*1*
Risk Rating:	***4***

The last, but not least, of the potential remote exploits uses the Simple Network Management Protocol (SNMP). In Chapter 3, we touched on how SNMP can be used to enumerate information on Windows NT systems running SNMP agents configured with default community strings such as "public." Windows 9*x* will spill similar information if the SNMP agent is installed (from the \tools\reskit\netadmin\snmp directory on Windows 9*x* media). Unlike NT, however, Windows 9*x* does not include Windows-specific information such as user accounts and shares in its SNMP version 1 MIB. Opportunities for exploitation are limited via this avenue.

Windows 9*x* Backdoor Servers and Trojans

Assuming that file sharing, the Dial-Up Server, and remote Registry access aren't enabled on your Windows 9*x* system, can you consider yourself safe? The answer to this question should be obvious by now—no. If intruders are stymied by the lack of remote administration tools for their target system, they will simply attempt to install some.

Next, we discuss three of the most popular backdoor client/server programs circulating the Internet. We also discuss the typical delivery vehicle of a back door, the *Trojan horse,* a program that purports to be a useful tool but that actually installs malicious or damaging software behind the scenes. Of course, scores of such tools are available on the Net, and there aren't nearly enough pages here to catalog them all. Some good places to find more information about back doors and Trojan horses are TLSecurity at http://www.tlsecurity.net/ and http://www.eqla.demon.co.uk/trojanhorses.html.

Back Orifice

Popularity:	*10*
Simplicity:	*9*
Impact:	*10*
Risk Rating:	***10***

One of the most celebrated Windows 9*x* hacking tools to date, Back Orifice (BO) is billed by its creators as a remote Windows 9*x* administration tool. Back Orifice was released in the summer of 1998 at the Black Hat security convention (see http://www

.blackhat.com/) and is still available for free download from http://www.cultdeadcow.com/tools/. Back Orifice allows near-complete remote control of Windows 9*x* systems, including the ability to add and delete Registry keys, reboot the system, send and receive files, view cached passwords, spawn processes, and create file shares. Others have written plug-ins for the original BO server that connect to specific IRC (Internet Relay Chat) channels such as #BO_OWNED and announce a BO'ed machine's IP address to any opportunists frequenting that venue.

BO can be configured to install and run itself under any filename. ([space].exe is the default if no options are selected.) It will add an entry to `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices` so that it is restarted at every system boot. It listens on UDP port 31337 unless configured to do otherwise. (Guess what the norm is?)

Obviously, BO is a hacker's dream come true, if not for meaningful exploitation, at least for pure malfeasance. BO's appeal was so great that a second version was released one year after the first: Back Orifice 2000 (BO2K, http://sourceforge.net/projects/bo2k/). BO2K has all the capabilities of the original, with two notable exceptions: (1) both the server and client run on Windows NT/2000 (not just Windows 9*x*), and (2) a developer's kit is available, making custom variations extremely difficult to detect. The default configuration for BO2K is to listen on TCP port 54320 or UDP 54321 and to copy itself to a file called UMGR32.EXE in %systemroot%. It will disguise itself in the task list as EXPLORER to dissuade forced shutdown attempts. If deployed in Stealth mode, it will install itself as a service called "Remote Administration Service" under the Registry key `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServices` that will launch at startup and delete the original file. All these values are trivially altered using the bo2kcfg.exe utility that ships with the program. Figure 4-5 shows the client piece of BO2K, bo2kgui.exe, controlling a Win98SE system. Incidentally, Figure 4-5 shows that now the BO2K client can actually be used to stop and remove the remote server from an infected system, using the Server Control | Shutdown Server | DELETE option.

TIP A lightly documented feature of the BO2K client is that it sometimes requires you to specify the port number in the Server Address field (for example, 192.168.2.78:54321 instead of just the IP or DNS address).

Popularity:	*8*
Simplicity:	*9*
Impact:	*8*
Risk Rating:	***8***

Similar in function but unrelated to BO, NetBus can also be used to take control of remote Windows systems (including Windows NT/2000). Written by Carl-Fredrik Neikter,

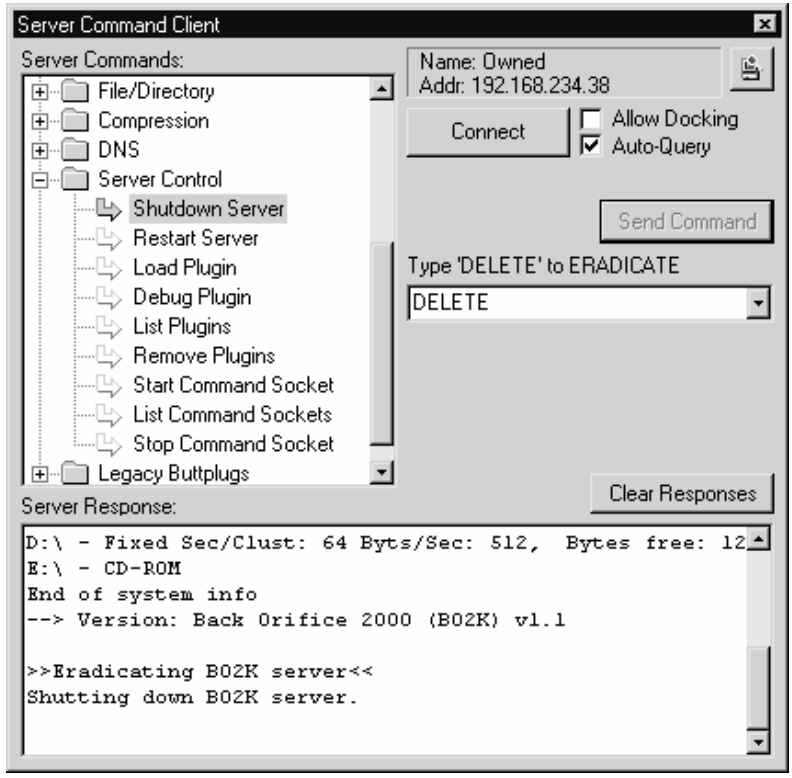


Figure 4-5. The Back Orifice 2000 (BO2K) client GUI (bo2kgui.exe) controlling a back-doored Win9*x* system. This is the way to remove the BO2K server.

NetBus offers a slicker and less cryptic interface than the original BO, as well as more effective functions such as graphical remote control (only for fast connections). NetBus is also quite configurable, and several variations exist among the versions circulating on the Internet. The default server executable is called patch.exe (but can be renamed to anything), which is typically written to `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run` so that the server is restarted every time the system boots. NetBus listens on TCP port 12345 or 20034 by default (also completely configurable). Because it cannot use UDP (like BO2K), it is more likely to get screened out at firewalls.

SubSeven

Popularity:	*10*
Simplicity:	*9*
Impact:	*10*
Risk Rating:	***10***

Judging by the frequency with which the authors' systems are scanned for this backdoor server, SubSeven has easily overtaken BO, BO2K, and NetBus combined in popularity. It certainly is more stable, easier to use, and offers greater functionality to attackers than the other three. It is available from http://come.to/subseven.

The SubSevenServer (S7S) listens to TCP port 27374 by default, and that is the default port for client connections as well. Like BO and NetBus, S7S gives the intruder fairly complete control over the victim's machine, including the following:

- Launching port scans (from the victim's system!)
- Starting an FTP server rooted at C:\ (full read/write)
- Remote Registry editor
- Retrieving cached, RAS, ICQ, and other application passwords
- Application and port redirection
- Printing
- Restarting the remote system (cleanly or forced)
- Keystroke logger (listens on port 2773 by default)
- Remote terminal (The Matrix; listens on port 7215 by default)
- Hijacking the mouse
- Remote application spying on ICQ, AOL Instant Messenger, MSN Messenger, and Yahoo! Messenger (default port 54283)
- Opening a web browser and going to a user-defined site

The server also has an optional IRC connection feature, which the attacker can use to specify an IRC server and the channel the server should connect to. The S7S then sends data about its location (IP address, listening port, and password) to participants in the channel. It also can act as a standard IRC robot (or "bot"), issuing channel commands, and so on. In addition, S7S can notify attackers of successful compromises via ICQ and e-mail.

Using the EditServer application that comes with S7S, the server can be configured to start at boot time by placing an entry called "WinLoader" in the Run or RunServices Registry keys or by writing to the WIN.INI file.

In a post to a popular Internet security mailing list, a representative of a major U.S. telecommunications company complained that the company's network had been inundated with S7S infections affecting a large number of machines between late January and early March of 2000. All these servers connected to a "generic" IRC server (that is, irc .ircnetwork.net, rather than a specific server) and joined the same channel. They would send their IP address, listening port, and password to the channel at roughly five-minute intervals. As the final sentence of the post read, "With the server putting its password information in an open channel, it would be possible for anyone in the channel with the Sub7 client to connect to the infected machines and do what they will." Without a doubt, SubSeven is a sophisticated and insidious network attack tool. Its remote FTP server option is shown in Figure 4-6.

Backdoor Countermeasures

All these backdoor servers must be executed on the target machine. They cannot be launched from a remote location (unless the attacker already owns the system, of course). This is typically accomplished by exploiting known flaws in Internet clients and/or just plain trickery. Wily attackers will probably use both. These methods are discussed at length in Chapter 16, "Hacking the Internet User," where countermeasures are also discussed. Here's a sneak preview: Keep your Internet client software up-to-date and conservatively configured.

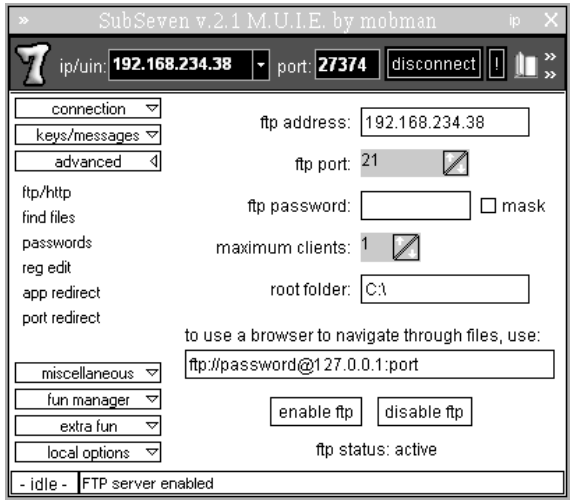


Figure 4-6. The SubSeven client enables an FTP server on the remote victim's system.

Another good way to block back doors is to prevent inbound access to listening ports commonly used by such programs. Many sites we've come across allow high ports over the firewall, making it child's play to connect to listening backdoor servers on internal networks. A comprehensive list of backdoor and Trojan ports is available on the excellent TLSecurity site at http://www.tlsecurity.net/trojanh.htm.

Pay close attention to outbound firewall access control as well. Although smarter attackers will probably configure their servers to communicate over ports such as 80 and 25 (which are almost always allowed outbound), it nevertheless helps to minimize the spectrum available to them.

If you get caught anyway, let's talk about fixing backdoor servers. For those with an inclination to go digging for the root of a problem so that they can ensure that it is manually pulled out, check out the excellent and comprehensive TLSecurity Removal Database at http://www.tlsecurity.net/tlfaq.htm. This page's author, Int_13h, has performed yeoman's work in assembling comprehensive and detailed information on where these tools hide. (Is it possible he's covered *every* known back door and Trojan horse? What a list!)

For those who just want to run a tool and be done with it, many of the major antivirus software vendors now scan for all these tools. (For a good list of commercial vendors, search for Microsoft's Knowledge Base Article Q49500 at http://search.support.microsoft .com.) Int_13h highly recommends Vexira, formerly the AntiViral Toolkit Pro (AVP), available at http://www.centralcommand.com/. A number of companies offer tools specifically targeted at removal of back doors and Trojan horses, such as the Trojan Defense Suite (TDS) at http://www.multimania.com/ilikeit/tds2.htm (another Int_13h recommendation).

Beware of wolves in sheep's clothing. For example, one BO-removal tool called BoSniffer is actually BO itself in disguise. Be skeptical about freeware Trojan horse cleaners in general.

We will further examine back doors and Trojan horses in Chapter 14.

Known Server Application Vulnerabilities

BO isn't the only piece of software that leaves the host system vulnerable to attack—plenty of commercial and noncommercial tools do this unintentionally. It would be nearly impossible to exhaustively catalog all the Windows 9*x* software that has had reported security problems, but there's an easy solution for this issue: Don't run server software on Windows 9*x* unless you really know how to secure it. One example of such a popular but potentially revealing server application is Microsoft's Personal Web Server. Unpatched versions can reveal file contents to attackers who know the file's location and request it via a nonstandard URL. (See http://www.microsoft.com/technet/security/bulletin/MS99-010.asp for more information.)

On a final note, we should emphasize that deploying "mainstream" remote control software such as pcAnywhere on a Windows 9*x* box throws all our previous advice out the window. If pcAnywhere is not properly configured, anyone can take over your system just as if they were sitting at the keyboard. We'll talk exclusively about remote control software in Chapter 13.

Windows 9*x* Denial of Service

Popularity:	*8*
Simplicity:	*9*
Impact:	*8*
Risk Rating:	*8*

Denial of service attacks are the last resort of a desperate mind. Unfortunately, they are a reality on the wild and wooly Internet. Numerous programs have the capability of sending pathologically constructed network packets to crash Windows 9*x*, with names such as ping of death, teardrop, land, and WinNuke. Although we talk in-depth about denial of service in Chapter 12, we will note the relevant patch for the Windows 95 versions of these bugs here: the Dial-Up Networking Update 1.3 (DUN 1.3).

Denial of Service Countermeasures

DUN 1.3 includes a replacement for the Windows 95 Windows Sockets (Winsock) software library that handles many of the TCP/IP issues exploited by these attacks. Windows 98 users do not need to apply this patch, unless they are North American users wanting to upgrade the default 40-bit encryption that comes with Windows 98 to the stronger 128-bit version. The Windows 95 DUN 1.3 patch can be found at http://www.microsoft.com/windows95/downloads/.

Even with the DUN 1.3 patch installed, we would advise strongly against deploying any Windows 9*x* system directly on the Internet (that is, without an intervening firewall or other security device).

Personal Firewalls

To top off our section on remote attacks, we strongly recommend purchasing one of the many personal firewall applications available today. These programs insert themselves between your computer and the network, and they block specified traffic. Our favorite is BlackICE PC Protection ($39.95 for a single user license at this writing), available from Internet Security Systems at http://blackice.iss.net/. Some other popular products we've seen in our travels include ZoneAlarm (free for home or nonprofit use, from Zone Labs at http://www.zonelabs.com/) and Aladdin's eSafe Desktop (see http://www.esafedesktop.com). For real peace of mind, obtain these tools and configure them in the most paranoid mode possible, as the default configurations often leave you with much less protection than you think.

WINDOWS 9*x* LOCAL EXPLOITS

It should be fairly well established that users would have to go out of their way to leave a Windows 9*x* system vulnerable to remote compromise. Unfortunately, the opposite is

true when the attackers have physical access to the system. Indeed, given enough time, poor supervision, and an unobstructed path to a back door, physical access typically results in bodily theft of the system. However, in this section, we will assume that wholesale removal of the target is not an option and highlight some subtle (and not so subtle) techniques for extracting critical information from Windows 9*x*.

Bypassing Windows 9*x* Security: Reboot!

Popularity:	*8*
Simplicity:	*10*
Impact:	*10*
Risk Rating:	***9***

Unlike Windows NT, Windows 9*x* has no concept of secure multiuser logon to the console. Therefore, anyone can approach Windows 9*x* and simply either power on the system or hard-reboot a system locked with a screensaver. Early versions of Windows 95 even allowed CTRL-ALT-DEL or ALT-TAB to defeat the screensaver! Any prompts for passwords during the ensuing boot process are purely cosmetic. The Windows password simply controls which user profile is active and doesn't secure any resources—other than the password list (see later in this chapter). It can be banished by clicking the Cancel button, and the system will continue to load normally, allowing near-complete access to system resources. The same goes for any network logon screens that appear. (They may vary depending on what type of network the target is attached to.)

Countermeasures for Console Hacking

One traditional solution to this problem is setting a BIOS password. The BIOS (Basic Input Output System) is hard-coded into the main system circuit board and provides the initial bootstrapping function for IBM-compatible PC hardware. It is therefore the first entity to access system resources, and almost all popular BIOS manufacturers provide password-locking functionality that can stop casual intruders cold. Truly dedicated attackers could, of course, remove the hard disk from the target machine and place it in another without a BIOS password. A few BIOS-cracking tools can also be found on the Internet, but BIOS passwords will deter most casual snoopers.

Of course, setting a screensaver password is also highly recommended. This is done via the Display Properties control panel's Screen Saver tab. One of the most annoying things about Windows 9*x* is that there is no built-in mechanism for manually enabling the screensaver. One trick we use is to employ the Office Startup Application (OSA), available when the Microsoft Office suite of productivity tools is installed. OSA's `-s` switch enables the screensaver, effectively locking the screen each time it is run. We like to put a shortcut to `osa.exe -s` in our Start menu so that is readily available. See Microsoft Knowledge Base (KB) article Q210875 for more information (http://support.microsoft.com/default.aspx?scid=kb;en-us;210875).

A few commercial Windows 9*x* security tools provide system-locking or disk-encryption facilities beyond the BIOS. The venerable Pretty Good Privacy (PGP), now commercialized but still free for personal use from http://www.pgp.com, provides public-key file encryption in a Windows version.

Autorun and Ripping the Screensaver Password

Popularity:	*4*
Simplicity:	*7*
Impact:	*10*
Risk Rating:	***7***

Hard rebooting or using the three-fingered salute (CTRL-ALT-DEL) to defeat security may offend the sensibilities of the elitist system cracker (or cautious system administrator who has forgotten their screensaver password), but fortunately there is a slicker way to defeat a screensaver-protected Windows 9*x* system. It takes advantage of two Windows 9*x* security weaknesses: the CD-ROM Autorun feature and poor encryption of the screensaver password in the Registry.

The CD-ROM Autorun issue is best explained in Microsoft Knowledge Base article Q141059:

> "Windows polls repeatedly to detect if a CD-ROM has been inserted. When a CD-ROM is detected, the volume is checked for an Autorun.inf file. If the volume contains an Autorun.inf file, programs listed on the 'open=' line in the file are run."

This feature can, of course, be exploited to run any program imaginable. (Back Orifice or NetBus, anyone?) But the important part here is that under Windows 9*x*, this program is executed even while the screensaver is running.

Enter weakness No. 2: Windows 9*x* stores the screensaver password under the Registry key `HKEY\Users\.Default\Control Panel\ScreenSave_Data`, and the mechanism by which it obfuscates the password has been broken. Therefore, it is a straightforward matter to pull this value from the Registry (if no user profiles are enabled, C:\Windows\ USER.DAT), decrypt it, and then feed the password to Windows 9*x* via the standard calls. Voilà—the screensaver vanishes!

A tool called SSBypass that will perform this trick is available from Amecisco for $39.95 (http://www.amecisco.com/ssbypass.htm). Stand-alone screensaver crackers also exist, such as 95sscrk, which can be found on Joe Peschel's excellent cracking tools page at http://users.aol.com/jpeschel/crack.htm, along with many other interesting tools. 95sscrk won't circumvent the screensaver, but it makes short work of ripping the screensaver password from the Registry and decrypting it:

```
C:\TEMP>95sscrk
Win95 Screen Saver Password Cracker v1.1 - Coded by Nobody (nobody@engelska.se)
(c) Copyrite 1997 Burnt Toad/AK Enterprises - read 95SSCRK.TXT before usage!
```

```
-------------------------------------------------------------------
· No filename in command line, using default! (C:\WINDOWS\USER.DAT)
· Raw registry file detected, ripping out strings...
· Scanning strings for password key...
 Found password data! Decrypting ... Password is GUESSME!
_ Cracking complete! Enjoy the passwords!
-------------------------------------------------------------------
```

Countermeasures: Shoring Up the Windows 9*x* Screensaver

Microsoft has a fix that handles the screensaver password in a much more secure fashion—it's called Windows NT/2000. But for those die-hard Win9*x* users who at least want to disable the CD-ROM Autorun feature, the following excerpt from Microsoft Knowledge Base Article Q126025 will do the trick:

1. In Control Panel, double-click System.
2. Click the Device Manager tab.
3. Double-click the CD-ROM branch and then double-click the CD-ROM driver entry.
4. On the Settings tab, click the Auto Insert Notification check box to clear it.
5. Click OK or Close until you return to Control Panel. When you are prompted to restart your computer, click Yes.

Revealing Windows 9*x* Passwords in Memory

Popularity:	*8*
Simplicity:	*9*
Impact:	*8*
Risk Rating:	***8***

Assuming that attackers have defeated the screensaver and have some time to spend, they could employ onscreen password-revealing tools to "unhide" other system passwords that are obscured by those pesky asterisks. These utilities are more of a convenience for forgetful users than they are attack tools, but they're so cool that we have to mention them here.

One of the most well-known password revealers is Revelation by SnadBoy Software (http://www.snadboy.com), shown working its magic in Figure 4-7.

Another great password revealer is ShoWin from Robin Keir at http://www.foundstone.com/rdlabs/tools.php?category=Forensic. Other password revealers include Unhide from Vitas Ramanchauskas (www.webdon.com), who also distributes

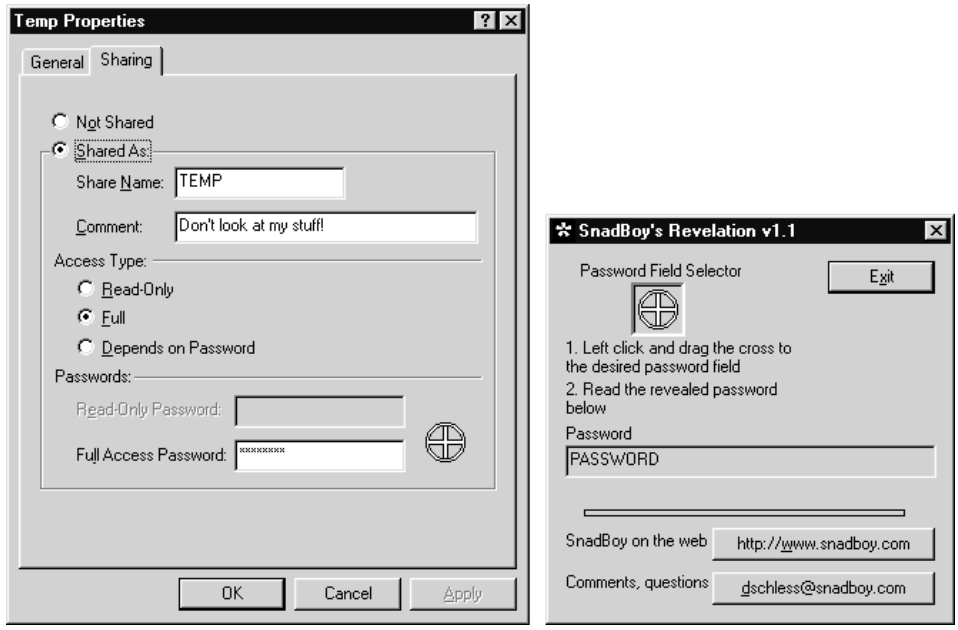


Figure 4-7. SnadBoy Software's Revelation 1.1 "unhides" a Windows file share password.

`pwltool` (see the next section) and the Dial-Up Ripper (dripper, from Korhan Kaya, available in many Internet archives), which performs this trick on every Dial-Up Networking connection with a saved password on the target system. Again, these tools are pretty tame considering that they can only be used during an active Windows logon session. (If someone gets this far, they've got access to most of your data anyway.) But these tools can lead to further troubles if someone has uninterrupted access to a large number of systems and a floppy disk containing a collection of tools such as Revelation. Just think of all the passwords that could be gathered in a short period by the lowly intern hired to troubleshoot your Win9*x* systems for the summer! Yes, Windows NT is also "vulnerable" to such tools—and, no, it doesn't work on network logon screens or on any other password dialog boxes where the password has not been saved. (That is, if you don't see those asterisks in the password box, you're out of luck.)

PWL Cracking

Popularity:	*8*
Simplicity:	*9*
Impact:	*8*
Risk Rating:	*8*

Attackers don't have to sit for long at a terminal to get what they want. They can dump required information to a floppy and decrypt it later at their leisure, in much the same way as the traditional UNIX crack and the Windows NT L0phtcrack password file–cracking approaches.

The encrypted Windows 9*x* password list, or .PWL file, is found in the system root directory (usually C:\Windows). These files are named for each user profile on the system, so a simple batch file on a floppy disk in drive A that executes the following will nab most of them:

```
copy C:\Windows\*.pwl a:
```

A .PWL file is really only a cached list of passwords used to access the following network resources:

- Resources protected by share-level security
- Applications that have been written to leverage the password-caching application programming interface (API), such as Dial-Up Networking
- Windows NT computers that do not participate in a domain
- Windows NT logon passwords that are not the Primary Network Logon
- NetWare servers

Before OEM System Release 2 (OSR2), Windows 95 used a weak encryption algorithm for .PWL files that was cracked relatively easily using widely distributed tools. OSR2 was an interim release of Windows 95 made available only through new systems purchased from original equipment manufacturers (OEMs)—that is, the company that built the system. The current PWL algorithm is stronger but is still based on the user's Windows logon credentials. This makes password-guessing attacks more time-consuming, but doable.

One such PWL-cracking tool is `pwltool` by Vitas Ramanchauskas and Eugene Korolev (see http://www.webdon.com). `Pwltool`, shown in Figure 4-8, can launch dictionary or brute-force attacks against a given .PWL file. Therefore, it's just a matter of dictionary size (`pwltool` requires wordlists to be converted to all uppercase) or CPU cycles before a .PWL file is cracked. Once again, this is more useful to forgetful Windows users than as a hacking tool. We can think of much better ways to spend time than cracking Windows 9*x* .PWL files. In the purest sense of the word, however, we still consider this a great Windows 9*x* hack.

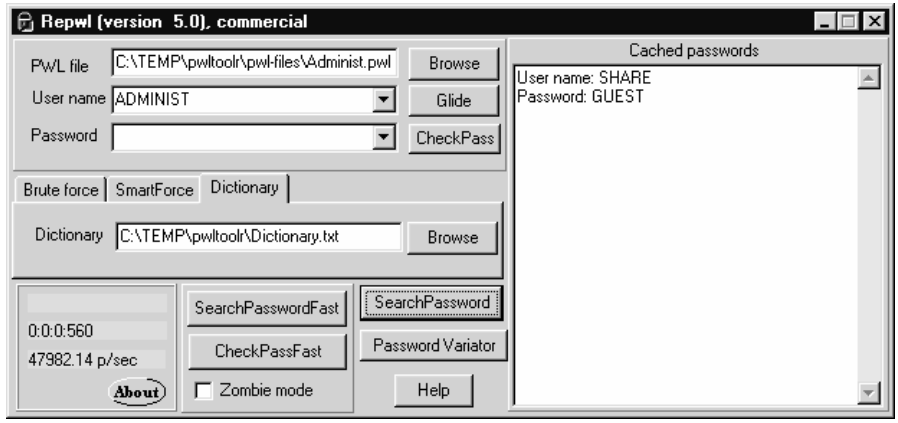


Figure 4-8. `Pwltool` unlocks the Windows 9*x* PWL password cache file.

Another good PWL cracker is CAIN by Break-Dance (see http://www.confine.com). PWL cracking isn't the only thing CAIN does, however. It will also rip the screensaver password from the Registry and enumerate local shares, cached passwords, and other system information.

Countermeasures: Protecting .PWL Files

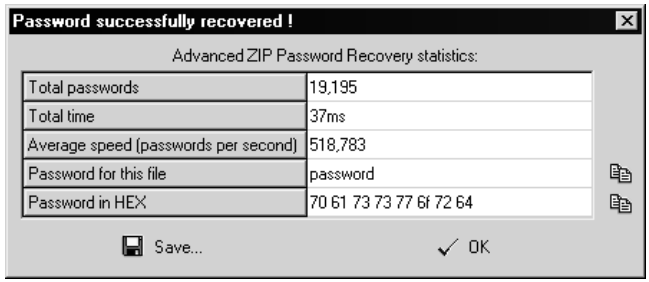
For administrators who are really concerned about this issue, the Windows 9*x* System Policy Editor can be used to disable password caching, or the following DWORD Registry key can be created/set:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\
      Network\DisablePwdCaching = 1
```

Those still using the pre-OSR2 version of Windows 95 can download the update to the stronger PWL encryption algorithm by following instructions at http://support.microsoft.com/support/kb/articles/Q132/8/07.asp.

PWL files aren't the only things the productivity-challenged programmers of the world have developed cracking tools for. The site at http://www.lostpassword.com lists utilities for busting everything from password-protected Microsoft Outlook .PST files to Microsoft Word, Excel, and PowerPoint files. (Who do you want to crack today?) There are even several crackers available for the ubiquitous .ZIP files that so many rely on to password-protect sensitive files sent over the Internet. Elcomsoft's Advanced Zip Password Recovery (AZPR) is capable of dictionary, plaintext, and brute-force cracks. Best of all,

it's incredibly fast, as illustrated in the following screenshot showing the results of a ZIP cracking session that burned along at an average 518,783 password guesses per second:



Another good site for password testing and recovery tools is Joe Peschel's resource page at http://users.aol.com/jpeschel/crack.htm. It's nice to know that whatever mess passwords can get you into can be reversed by your friendly neighborhood hacker, isn't it?

WINDOWS MILLENNIUM EDITION (ME)

Windows Millennium Edition (Windows Me) is a "refreshed" version of Windows 98 with some fixes and new usability features.

Windows Me Remote Attacks

From a remote attacker's perspective, Windows Me looks just about like Windows 9*x* does. No new services have been introduced. File and print sharing are disabled by default, as is the Remote Registry Service. Unless the end user turns something on, remote penetration of Windows Me is highly improbable using published techniques.

Windows Me Local Attacks

In terms of local attacks, Windows Me is also pretty much identical to Windows 9*x*. One of the most interesting recent attacks unique to Windows Me (and Windows 98 with the Plus! add-on package installed) involved a longstanding security issue that we are frequently asked about in our consulting work: How do you protect specific files from multiple users of a Win9*x*/Me system? The standard scenario is a small office / home office (SOHO) environment that doesn't need or want to expand the number of machines used for specific purposes. Therefore, it uses a single machine for multiple roles. For example, consider a dentist's office where the receptionist uses a Windows Me system to manage patient schedules during the day, while the same machine also hosts the accounting application used during the evenings by the office manager. How do you keep the receptionist from examining the office's financial records?

As you might imagine, the default solution to this problem is to use features built in to the operating system, whether they actually provide a true remedy or not.

Recovering Passwords from Compressed Folders

Popularity:	*8*
Simplicity:	*9*
Impact:	*8*
Risk Rating:	*8*

Windows 98 Plus! and Me provide a feature called Compressed Folders that transparently compresses any files moved into such a folder to save space on the disk. Microsoft provides a password-protection feature with Compressed Folders that gives Windows 98 Plus! and Me users the impression that Compressed Folders can be used to password-protect files stored in Compressed Folders. We have seen many small businesses utilize this feature in scenarios like those mentioned earlier to protect sensitive files from specific users of the same Windows 98 Plus! or Me machine. Unfortunately, this feature does not provide the type of security that these small businesses assume it does.

The root of this vulnerability is that Compressed Folder passwords are logged in cleartext to the file c:\windows\dynazip.log on the local Windows 98 Plus! or Me system. Anyone who knows this can simply open up the file and view the password for any compressed folder on the system.

Countermeasures for Compressed Folder Password Recovery

The best response to this problem is to not rely on Compressed Folder passwords for security. Microsoft recommends upgrading to Windows NT or 2000 and using separate user accounts and NTFS access control lists to prevent multiple users from accessing each other's files.

NOTE We do not recommend using the Windows 2000 Encrypting File System for this role because it adds little additional authorization security to standard NTFS protections and is easily circumvented by a dedicated attacker with physical access to the system anyway.

Several third-party file security products are available for Windows 9*x* and Me. We recommend using these if you are not able or willing to take the step up to Windows NT or 2000. One of our favorites is PGP Disk from PGP Corporation (http://www.pgp.com). A list of freeware and demo Win9*x* file encryption tools is available at http://www.modemspeedtest.com/crypto.htm. We have not tested nor do we recommend any specific tools in this space, so be sure to evaluate them carefully before relying on them for mission-critical file protection.

For those who care to install patches on Microsoft features that shouldn't be used anyway, a fix is available from http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-019.asp. Make sure that if you apply this patch you also delete any *existing* c:\windows\dynazip.log files that are not erased by the patch. This vulnerability is associated with Bugtraq Vulnerability ID 2516.

SUMMARY

As time marches on, Win9*x*/Me will become less and less interesting to attackers as most potential victims move to newer OSs based on the much more stable and secure Windows NT Family codebase. For those users who remain stuck in the tar pits, take the following to heart:

- ▼ Windows 9*x*/Me is relatively inert from a network-based attacker's perspective because of its lack of built-in remote logon facilities. About the only real threats to Win9*x*/Me network integrity are file sharing, which can be fairly well secured with proper password selection, and denial of service, which is mostly addressed by the Dial-Up Networking Update 1.3 and Windows Me. Nevertheless, we strongly recommend against deploying unprotected Win9*x*/Me systems on the Internet. Combining the ease with which services can be enabled by unwary users and the lack of secondary defense mechanisms is a sure recipe for problems.
- ■ The freely available backdoor server tools such as SubSeven and several commercial versions of remote control software (see Chapter 13) can more than make up for Win9*x*/Me's lack of network friendliness. Make sure that neither is installed on your machine without your knowledge (via known Internet client security bugs such as those discussed in Chapter 16) or without careful attention to secure configuration (read: "good password choice").
- ■ Keep up with software updates because they often contain critical security fixes to weaknesses that will leave gaping holes if not patched. For more information on the types of vulnerabilities unpatched software can lead to and how to fix them, see Chapter 16.
- ■ If someone attains physical access to your Win9*x* machine, you're dead in the water (as is true for most OSs). The only real solution to this problem is BIOS passwords and third-party security software.
- ▲ If you're into Windows 9*x* hacking just for the fun of it, we discussed plenty of tools to keep you busy, such as password revealers and various file crackers. Keep in mind that Windows 9*x* .PWL files can contain network user credentials, so network admins shouldn't dismiss these tools as too pedestrian, especially if the physical environment around their Windows 9*x* boxes is not secure.