

# Head Off Spyware, Viruses and Malware

**ARTICLE DATE:** 04.11.05

## **SPEC DATA**

*Editor's Note: PC Magazine has partnered with Wiley Books to create a series of PC Magazine and ExtremeTech books. In Fighting Spyware, Viruses and Malware technology expert and author Ed Tittel reveals the awful truth about most home PCs: they're riddled with spyware (and worse). Fortunately, he then tells readers what to do about it. This chapter explains how to keep the varmints off your PC in the first place.*

By Ed Tittel

To a large extent, practicing system safety is based on knowledge. That is, you have to know what's normal and regular on your system before you can recognize what's abnormal or irregular. In addition, there are plenty of commonsense rules of safe computing that are easy to practice given the installation, use, and proper maintenance of the right tools to help you stay safe, online and off. Throw in a few good working habits, and exercise a little due diligence before downloading software, and you're most of the way safe.

But as the old saying goes: "eternal vigilance is the price of liberty" (Wendell Phillips, 1852). Practicing safe computing means keeping up with current threats and vulnerabilities, so as to be better prepared to encounter and foil them. It also means keeping a constant eye on your system and its behavior to look for the odd, the unwanted, the unexpected, or other signs that malign forces may be at work somewhere. This chapter explains how to recognize a clean and safe PC, and how to keep your PC secure and pristine!

## **Baselining Your System**

Once you've done the scanning necessary—for viruses, spyware, adware, and so forth—to make reasonably sure your PC isn't operating under a cloud of sorts, you can take a look around your system to see what's normal. Computer geeks sometimes call this kind of activity "baselining" a system, because it's intended to provide you with a snapshot of what's normal for your PC.

You'll find several components of interest when establishing a baseline for your PC. One of the most important components involves taking a look at what processes are running and active on your system right after startup, before you fire off any applications. That way, they'll include only those processes that Windows itself launches at startup to do its job, and those associated with other programs that normally launch during startup (many items in this latter category will be related to the firewalls, anti-virus, anti-spyware, and other security or safety components I recommend elsewhere in this book, in fact). In the sections that follow, I describe some methods for taking snapshots of a normal baseline, including process and file inventories.

## **Creating a Process Inventory**

It's easy to take a process inventory at any time on a Windows XP machine. Simply right-click the task bar (usually at the bottom of your display area unless you've moved it) and select Task Manager from the resulting pop-up menu. To see the list of processes running on your PC, click the Processes tab at the top of the window, as shown in Figure 12-1.

### **Tip**

By clicking the heading button in any column on the Task Manager Processes tab, you can cause the processes to be sorted on that field value. Thus, for example, click CPU Time to sort processes according to the amount of CPU time they've consumed since the last reboot (hint: System Idle Process

Of course, unless you want to copy all this information by hand, it might be more sensible to record it to a file. That way, you've got something to compare things with later when you go back to check this list. Because you're trying to compare current conditions to what passes for normal on your PC, I recommend you do this right after startup, before launching any additional applications. Fortunately, the built-in Windows tasklist command makes this job easy; here's how you can ensconce this data safely in a file named tasklist-yymmdd.txt (substitute two-digit codes for the year, month, and day so you can tell when the snapshot was taken):

1. Open a Command Prompt window by choosing Start@>Run, typing cmd (or cmd.exe, if you prefer) in the Open text field of the Run dialog box, and then clicking OK to execute those instructions.
2. To see a list of active tasks on your system, type tasklist at the command prompt. This produces a display like that shown in Figure 12-2. For more information on the tasklist command, type tasklist /? to display online command help. Also, you can sort the process names alphabetically by typing tasklist /nh | sort (this drops the column headings from the output and then sends the resulting process names and info to a sort utility to sort them in descending alphabetical order by process name).
3. At the command prompt, type tasklist > C:\tasklist-yymmdd.txt, where you substitute two-digit values for year for yy, month for mm, and day for dd.

always wins, so look in line 2 and lower for potential causes for concern). Click again on the same heading to reverse the sort order (by default, the highest values show up first, so clicking again causes the lowest values to show up first). Same thing goes for Mem Usage (memory usage), which can also be pretty revealing when it comes to understanding where your system resources are going.

**Note:** Indeed you could cut and paste the text directly from the command window that's shown in Figure 12-2, but I go on to Step 3 and have you repeat the command, piping the output directly into a file. I think it's an easier and more straightforward way to grab this information and put it some place you can find it again. The syntax I show for the command writes the output to the root of the C:\ drive. If you follow those instructions verbatim, you may want to move that file somewhere else so you can find it more easily at another time.

4. Type exit at the command line or click the x-shaped close control in the upper right-hand corner of the command window to close this window.

## Understanding What You See

For each entry under the Image Name heading (appears in both the Task Manager Processes display and in tasklist command output), you determine what it represents and whether it's benign (as it should be) or malign (which means it needs rooting out). Let's look at a reformatted version of the tasklist output in Table 12-1 (entries were alphabetized and some column data removed), which ties into the 16-item numbered list shown after Table 12-1.

Image Name	PID	Mem Usage	Explanation
alg.exe	124	2,584K	Application Layer Gateway (1)
CCAPP.EXE	1296	23,892K	Common Client application (NAV; 2)
CCEVTMGR.EXE	1632	3,732K	Symantec Common Client Event Mgr Svc (NAV; 3)
CCPROXY.EXE	1912	6,076K	Symantec Common Client Proxy Server (4)
CCSETMGR.EXE	1608	4,312K	Background task associated with NIS (5)
cmd.exe	2884	872K	Windows command shell (Command prompt window)
csrss.exe	928	816K	Windows client server runtime subsystem (6)
explorer.exe	1520	19,180K	Windows Program Manager a.k.a. Windows Explorer
explorer.exe	2784	2,696K	Child process of Windows Program Manager
hh.exe	2556	16,748K	Windows help program
iexplore.exe	2776	22,480K	Microsoft Internet Explorer (7)
lsass.exe	1020	1,348K	Windows Local Security Authority Service (8)
mgabg.exe	1960	1,840K	Matrox BIOS Guard (works with graphics card)
msmsgs.exe	748	2,944K	MSN Messenger Traybar service (9)
NAVAPSV.C	1992	8,220K	Norton AntiVirus auto-protect service (10)
NOPDB.EXE	500	10,096K	Norton Speed Disk process (11)

NPROTECT.EXE	236	5,284K	Norton process protects recycle bin (12)
pdesk.exe	552	3,620K	System tray app for Matrox graphics card
realsched.exe	2896	176K	Scheduler program: prompts for RealOne updates
SAVSCAN.EXE	300	616K	Symantec's anti-virus scanning software
services.exe	1008	3,504K	Used to start, stop, and manage system services
smss.exe	404	408K	Session Manager Subsystem (13)
SNDSrv.exe	464	3,360K	Symantec Network Driver Svc (part of NIS2003/4)
spoolsv.exe	1788	5,036K	Windows print spooler service (14)
SpySweeper.exe	1284	10,196K	SpySweeper anti-spyware/anti-adware background task
svchost.exe	624	3,272K	Windows system process to service dynamic link libraries (DLLs) (15)
svchost.exe	1180	4,772K	Windows system process to service DLLs (15)
svchost.exe	1228	3,688K	Windows system process to service DLLs (15)
svchost.exe	1308	27,376K	Windows system process to service DLLs (15)
svchost.exe	1356	2,440K	Windows system process to service DLLs (15)
svchost.exe	1408	3,484K	Windows system process to service DLLs (15)
symclsvc.exe	532	520K	Symantec Core Library Code (common code items)
System	4	224K	The Windows System process
System Idle Process	0	16K	Runs whenever CPU is idle
ups.exe	572	1,788K	PowerChute uninterruptible power supply (UPS) monitoring tool
winlogon.exe	964	7,964K	Windows process to manage user logon and logoff
wmiprvse.exe	3924	4,560K	Windows Management Interface (WMI) provider service (16)

1. The Application Layer Gateway is a Microsoft executable that provides functionality for the Windows Firewall and for Internet Connection Sharing for Windows XP. I find no evidence to indicate this process may be impersonated or subverted by spyware, adware, or malware, but many attacks that attempt to shut down local security will attempt to shut down this process as part of that effort.

**Note:** All of the executables that start with CC are part of the Symantec Common Client runtime environment, used for Norton Internet Security (which includes Norton Personal Firewall, Norton AntiVirus, Norton AntiSpam, and various other components in the test installation). This includes entries 2, 3, 4, and 5 on this list. There are no known attacks that impersonate these Norton components, as best I can discover.

2. CCAPP.EXE is part of the Norton AntiVirus system. No documented attacks or impersonations on many of these components, though some malware may attempt to shut down one or more of these components to bring down Norton security shields.

3. CCEVTMGR.EXE provides a general event management registration and reporting service for all Norton Internet Security components.

4. CCPROXY.EXE provides a mechanism for proxying Web access requests within Windows environments where the Norton Personal Firewall is active; it's designed to let the software screen outgoing Web requests according to security and suitability criteria (the latter in connection with Norton Internet Security's Parental Controls).

5. CCSETMGR.EXE provides a mechanism for launching various Norton Internet Security or Norton AntiVirus components at startup, and for scheduling LiveUpdate automated downloads.

6. cmd.exe is the Windows executable for the command-line environment (this appears only if you have a command prompt window open when the snapshot is taken). No known impersonation or attacks are documented, though some malware may open this process to handle scripts if system security is sufficiently compromised.

7. csrss.exe is the Windows client server runtime subsystem. Its job is to provide common windows, thread management, and graphics capabilities to all subsystems running in the Windows environment.

**CAUTION:** At least one known virus impersonates csrss.exe, so be very suspicious if you see more than one instance with this name (there should be only one).

8. lsass.exe is the Windows local security authority subsystem service that handles the logon process, user authentication, and generates session-specific security tokens that are compared with user and group permissions to determine whether resource access requests are granted or denied. The Sasser worm specifically attacks this system component, as do some varieties of Nimos and Lovgate.

9. msmsgs.exe is what the MSN messenger service users to advertise its presence, and to include a traybar icon on the Windows XP desktop (installed by default in Windows XP and subsequent service packs). Although no attacks on this component are documented, some attacks use file transfers inside the application to try to deliver infected payloads to users.

**CAUTION:** Several documented viruses use msmsgs.exe as their process names. You should never see more than one instance of this process name (or even one if you disable the Windows Messenger application). If you don't use Windows Messenger, in fact, it's perfectly safe to terminate this process. If you don't use Messenger at all, or if you don't mind starting it up manually yourself (use the Run command, type msmsgs.exe in the Open: text box, then click OK), you can stop it from running on startup as follows: Start@ @>All Programs@ @>Windows Messenger@ @>Tools@ @>Preferences and then uncheck the check box that reads "Run Windows Messenger when Windows starts."

10. NAVAPSV.C.EXE is Norton AntiVirus's auto-protect service; its job is to screen inbound and outbound file transfers, e-mail attachments, and so forth to block viruses from entering or leaving your PC. No known attacks are documented, but this is clearly something many types of malware will try to turn off if possible.

11. NOPDB.EXE is associated with the Norton Speed Disk utility in Norton SystemWorks; its job is to permit Speed Disk to launch during startup when the user requests this service. No known attacks are documented, nor any attempts to turn off or defeat this service. You won't see this on your machine unless it's also running Norton SystemWorks.

12. NPROTECT.EXE is associated with the Norton Protected Recycle Bin set up as part of Norton SystemWorks; its job is to prevent the Recycle Bin from being emptied without obtaining user confirmation. No known attacks are documented, nor any attempts to turn off or defeat this service.

13. smss.exe is a Microsoft process involved with creating, managing, and deleting user sessions.

**CAUTION:** Numerous viruses run using the smss.exe image name, so be sure to preserve only that version that resides inside the C:\Windows\System32 folder (all others are illegitimate).

14. spoolsv.exe is the Microsoft print spooler service, which stores pending print jobs on your PC until they can be sent to the designated printer for output.

**CAUTION:** Numerous viruses run using the spoolsv.exe image name, so preserve only that version that resides inside the C:\Windows\System32 folder (you can even end this process, too—you just won't be able to print unless you manually restart the Printer Spooler service or reboot your machine).

15. svchost.exe runs as a process that supports common Windows dynamic link libraries (DLLs) for lots of services. In fact, you'll see one instance of this executable in the processes display for each such group of services that shares a common set of DLLs in the Windows runtime environment. Figure 12-3 shows tasklist output that's been crafted to document what services are involved in each of the six svchost.exe instances present in svchost that appear therein—notice the wide variety and large number of services involved. No known attacks or attempts to turn off this process are documented, but it too should be found only resident in the C:\Windows\System32 folder (though you will find copies in service pack or CD image folders as well).

**Note:** The precise command syntax in Figure 12-3 is `tasklist /svc /fi "imagename eq svchost.exe"`. Restated in something closer to English, this means show me all the DLLs for the services that every instance of the svchost.exe file uses. What you see in that display are various instances of svchost, where the first relates to distributed communications and terminal

services, the second to remote procedure call services, the third to a whole bunch of services that call common presentation DLLs, the fourth to DNS caching behavior, the fifth to various kinds of remote access, and the sixth to a still digital imaging service that runs in Windows XP.

16. `wmiprvse.exe` is a manifestation of Microsoft's system management application program interfaces (APIs) at the runtime level; it's a rearchitected version of the WMI interface introduced in Windows XP (and also supported in Windows Server 2003) to support all WMI services through a single provider service. That's why you'll find it running on your system someday because you've recently installed an application that draws on WMI support, or a Microsoft update that does likewise. As with `svchost.exe`, you will occasionally encounter multiple instances of this software running at the same time—this is normal.

**CAUTION:** Some viruses that impersonate this service have been reported—most notably `Sonebot.B`, `Gletta.A.Trojan`, and various flavors of `Sasser`. The only valid instance of this code resides within the `C:\Windows\System32\wbem` directory.

By getting a sense of what's normal for your system, you can use Task Manager or the `tasklist` command at any time you've got reason to be concerned about your system to check one snapshot against the other. If you do your homework on the initial snapshot, you'll need to check up only on new items to figure out where they're coming from, and what kinds of trouble they might portend, if any.

This probably leads to a perfectly valid question: "How do I find out about process names on my PC?" Indeed, my own listings here are examples that will contain some (but not all) of the items that will show up on your machine. To document your unique collection of processes, open Task Manager and check the Processes tab, or create your own `tasklist` output file as described earlier. Then, you can use the various entries in the Image Name column on Google, Yahoo!, or the search engine of your choice to learn more about these processes—especially, whether they should be causes for concern or otherwise. I got my information from a whole variety of sources online, along with an excellent tool from Liutilities called `WinTasks 5 Professional` (see the "Resources" section toward the end of the chapter for more details on this offering).

**Note:** To get a definite sense of what "Safe mode" really means for Windows XP, try booting your machine into that mode (hold the F8 key down just as Windows starts booting and then select Boot in Safe mode from the resulting character mode menu that appears on your screen). Whereas my normal Windows XP boot shows 30-plus processes, in Safe mode I get only 12 (not counting `taskmgr.exe`, which runs only to show me the other process names), and those include only basic system elements necessary for operation: `csrss.exe`, `explorer.exe`, `lsass.exe`, `services.exe`, `smss.exe`, System, System Idle Process, and `winlogon.exe`, plus "only" three instances of `svchost.exe`! This really shows how few elements are needed to run a minimal, stripped-down version of Windows.

## Rough and Ready Performance Metrics

Although Windows XP does include a marvelous utility called Performance (it's in the Administrative Tools folder in the Control Panel) that you can use to measure system performance very accurately, you don't really need that tool to get a sense of what's normal on your PC. Instead, you can use a watch with a second hand because pinpoint accuracy isn't really overwhelmingly important (hence the title for this section).

Instead, create a text file or take notes with results from timing typical activities on your machine. They should include some or all of the following items:

- Normal startup time (cold boot)—Start timing as soon as you turn on the power to your PC and stop when the Windows login prompt appears (if applicable), or when the booting process has completed (if not).
- Normal restart time (warm boot)—Restart Windows XP (Start@ @>Turn Off Computer and then click the Restart button) and start timing simultaneously; stop timing when the Windows login prompt appears (if applicable), or when the booting process has completed (if not).
- Start time for commonly used applications—These might include Office components, Internet Explorer (or whatever Web browser you use), and other applications that take at least a short time to launch (to give you enough time to have something to measure). Launch them from the Start@ @>All Programs menu sequence and start timing as you click the application name on its pop-up menu. Stop timing when the application is ready for your input.

By comparing your baseline timings with those taken at another time, you'll be able to tell if your machine is running more slowly than usual or not.

## Other Snapshots Worth Gathering

Most professionals who go looking for signs of unwanted or malicious activity also depend on comparing before and after snapshots of key directories in the Windows file system and in the Windows registry. I touched on some of the techniques and tools involved in Chapter 4 of this book. If you decide you might want to use them on your own system, you'll need to get familiar with some new tools and techniques yourself.

The Windows directories where untoward things often happen include the %windir% directory (this environment variable usually points to C:\Windows on most Windows XP computers, but to C:\WINNT on Windows NT and 2000) and the %windir%\System32 subdirectory (a.k.a. C:\Windows\System32). By monitoring the contents of these directories, you can sometimes discover signs of unwanted software at work. By following the same steps to create a baseline snapshot now, and a comparison snapshot later, you can create a basis for investigation and see what's changed. Here's how:

1. Open a Command prompt window (Start@>Run, type cmd.exe in the Open text field of the Run dialog box, and then click OK).
2. Type the following at the command line: `dir %windir% /o:-d > winfiles-yymmdd.txt`, where yy is the two-digit year, mm the two-digit month, and dd the two-digit day. Note that this captures only the files in this directory (you'd use the /s -d attribute instead of /a: -d to capture subdirectory data as well).
3. Type the following at the command line: `dir %windir%\System32 /s /o:-d > win32files-yymmdd.txt`, where yy is the two-digit year, mm the two-digit month, and dd the two-digit day. Note that this captures all the files in the . . . \System 32 directory and all of its subdirectories, so this can be a big file.

### Tip

Finding differences isn't necessarily a bad thing-especially if you've installed a security update or a service pack since the last snapshot (in that case, you should expect to see things change so much that you'll really want to create a new baseline after performing such actions). The same thing applies whenever you install new or update existing software as well: make a new baseline! In general, it's only when you find instances of familiar file names in directories where they're not supposed to be (or in new directories for which you have no idea where they came from.) that you really

### Tip

If you want to be sure you're comparing "after" snapshots to current known good working "before" snapshots, it's essential to rebuild your baseline snapshots every time you change something about your PC. This means after installing new applications or utilities, service packs or security updates; adding new (or removing old) hardware; and so on. All of these things change the Windows registry, file system, and the list of processes active on your PC. Without keeping up with changes, you may end up chasing phantoms instead of

If you use this same process for your baseline and then when you're conducting an investigation, compare the various files for the different dates involved, and you may be able to spot some differences. Files will be listed newest first, so hopefully, you won't have to look too deeply into any list to see new or unexpected items in the "after" snapshots that are missing from the "before" snapshots.

You can also apply the same technique to your Windows registry, but it takes a bit more effort. The idea is to snapshot and export the contents of major registry keys (HKEY\_CLASSES\_ROOT or HKCR, HKEY\_CURRENT\_USER or HKCU, HKEY\_LOCAL\_MACHINE or HKLM, and so forth) or subkeys subject to change—for example, the HKLM\SOFTWARE key is the item to grab for before and after snapshots when installing software on your PC—to provide a basis for comparison.

If you don't want to spring for one of the tools I recommend in Chapter 4 (such as Registry Watch or Active Registry Monitor, which can perform such comparisons for you more or less automatically), you'll have to do a certain amount of setup and legwork to implement my suggestions (see also the next section, which specifically addresses issues involved in comparing snapshots to one another). Here's how to snapshot your major registry keys:

1. Launch the Windows Registry Editor: Start@ @>Run, type regedit.exe in the Open dialog box, and then click OK.
  2. Highlight the first major key in the registry, HKEY\_CLASSES\_ROOT, as shown in Figure 12-4.
  3. Click File and then Export in the resulting pull-down menu. The Export Registry File window appears, as depicted in Figure 12-5. Notice the file naming convention I used: RegSnap-HKCR-yyymmdd.reg. This helps you to identify and reimport that data should you ever need to and provides the basis for automated comparisons explained in the next section.
- Note:** The Registry Editor saves exported files by default in .reg format. That's good, because if you want to read data exported from your registry, or want the ability to restore only specific, individual keys and values, stick with the default registry file type (.reg extension). You will find other sources that recommend that you save such snapshots in hive file format (which usually take the .hiv extension) but if you do so, please follow other instructions carefully, realizing that you won't be able to read the contents of those files (even using WinDiff they're pretty incomprehensible for the most part) and that you can import only entire hive files in one go. In addition to being human readable, you can also pick and choose the keys and/or values you want to import from .reg files into your registry, which makes them preferable for most uses, in my opinion.
4. By default, the file is saved in your My Documents folder, but you can navigate inside the My Computer or My Network controls to store its contents elsewhere. Click the Save button, and you're done.
  5. Repeat for the other major registry keys (HKCU, HKLM, and HKU—you don't need to capture HKCC because it's dynamically rebuilt each time Windows starts up).
  6. Close the Registry Editor (click the x in the upper right-hand corner, or use Registry@ @>Exit menu commands).

Here again, you'll need to repeat this exercise later, so you'll have "after" snapshots to compare to your original baselines.

## Comparing Differences

If you've a mind to avoid lots of reading and manual labor when comparing differences between one snapshot file and another, you're not alone. In fact, Microsoft includes a special tool called Windiff.exe that's designed to compare two versions of the same file (or two similar files, as will be the case here) to one another.

## INSTALLING WINDIFF

WinDiff isn't installed as part of Windows XP (or other Windows versions) by default. You have to load your Windows XP CD or your latest Service Pack CD and install it from there. Here's how:

real problems. Thus, it might be a good idea to get in the habit of building new baselines each time you make a system change, and at least once a month (perhaps on the same day of each month, driven by an Outlook reminder?) to be doubly darn sure you're working from the latest and greatest known good working baseline of your PC.

1. Insert the CD into your CD drive; the autorun program on the CD should launch the Windows XP install utility, as shown in Figure 12-6.
2. Click the "Perform additional tasks" link that appears in Figure 12-6 and then click the Browse this CD button (this produces the display shown in Figure 12-7).
3. Open the SUPPORT folder to access the setup utility for the Windows Support Tools, as shown in Figure 12-8.
4. Double-click SETUP.EXE to launch the Windows Support Tools installation wizard. It will lead you through the rest of the installation process. If you decide you don't want to install the complete collection of Windows Support Tools, you can elect to install Typical Tools (rather than the complete set, which also includes Optional Tools) because WinDiff is included in the former subset.
5. When the installation is finished, close all open windows and you'll be able to start using WinDiff.

## USING WINDIFF

Once you've installed WinDiff, it shows up by default in a directory named %ProgramFiles%\Support Tools (for most readers, this means C:\Program Files\Support Tools). Using it requires a little preparation and understanding, but it's really not that bad. Here's how:

1. To launch the program, double-click the entry named WinDiff (or WinDiff.exe) in the Support Tools directory. Alternatively, you can click the WinLogo key and R and then type "%programfiles%\Support Tools\WinDiff" into the Open text field of the Run dialog box (note: the quotes around the string are necessary because the file specification has blanks in it). Either way, you should see a display like the one shown in Figure 12-9.
2. Next, click the File command in the WinDiff toolbar menu. There, the first two commands are Compare files and Compare directories. This admittedly contrived example hinges on comparing two directory lists I made of some product keys I keep around, one before I went in and added a file and changed some values in another file, the other after making such changes. These two files appear side-by-side to show the raw data in the following code lines:

Volume in drive D is Data and Storage	Volume in drive D is Data and Storage
Volume Serial Number is 2803-B30D	Volume Serial Number is 2803-B30D
Directory of d:\Test040928	Directory of d:\Test040928
08/05/2004 05:07 PM 39 bitdefender-key.txt	08/05/2004 05:07 PM 39 bitdefender-key.txt
09/19/2004 06:33 PM 60 NAV2005upg-install-key.txt	09/19/2004 06:33 PM 60 NAV2005upg-install-key.txt
08/05/2004 04:40 PM 76 NIS2004-install-key.txt	08/05/2004 04:40 PM 76 NIS2004-install-key.txt
09/17/2004 09:50 PM 368 NIS2005-install-key.txt	09/17/2004 09:50 PM 368 NIS2005-install-key.txt
07/11/2004 04:41 PM 30 NortonInetSecurityKey.txt	09/28/2004 04:45 PM 400 NIS2006-install-key.txt
08/23/2004 06:25 PM 58 opera-7-regcode.txt	07/11/2004 04:41 PM 30 NortonInetSecurityKey.txt
06/15/2004 03:23 PM 28 spysweeperkey.txt	08/23/2004 06:25 PM 58 opera-7-regcode.txt
7 File(s) 659 bytes	09/28/2004 04:46 PM 54 spysweeperkey.txt
0 Dir(s) 100,493,692,928 bytes free	8 File(s) 1,085 bytes
	0 Dir(s) 100,493,692,928 bytes free

**Note:** I did take some liberties with these listings, including deleting unnecessary white space to fit it onto the page, and adding a blank line to the right-hand file listing to make the file count and free space lines match for both files.

3. If you click the Compare Files menu in WinDiff, the first window that pops open in response lets you pick the first file for comparison. Once you specify that file, a second window that lets you pick the second file pops up next. In my case, I

compared a couple of directory listings named keyfilev1.txt and keyfilev2.txt that I deposited into the My Documents folder. After making these selections, a display like the one shown in Figure 12-10 appears (an analysis of this display, which is the meat of this whole section appears after the final step in this step-by-step sequence that follows the figure).

4. Once you're finished with WinDiff, click the red x in the upper right-hand corner, or select File@@>Exit to close the application.

The top line of the WinDiff display area shows the two files being compared. In the actual listing, lines with differences between the two files show up in yellow for the file 2 information, red for the file 1 information. This means that a line that's present in file 2 but not in file 1 (an added line) shows up only in yellow. This is the case for the unnumbered line between line numbers 9 and 10, where the listing for file NIS2006-install-key.txt shows up. A line that was present in file 1 but absent in file 2 would show up in red only (this does not occur in this example). A line that differs between the two files shows up in red first for the file 1 version and yellow second for the file 2 version. This is the case for lines 12 and [12] and for lines 13 and the unnumbered line that follows immediately afterward. These pairs of lines show that the file size for spysweeperkey.txt changed from 28 in file 1 to 56 in file 2, and that the total byte count for file 1 is 659, but 1,085 for file 2. This is just the kind of information you need to compare directory contents, registry files, and other items that may have changed. The appearance of a new file (for a product as yet unannounced, and by no means available) is a sure sign of monkey business, as are the changes in file size for the Spy Sweeper key and for the directory itself.

Working with WinDiff takes a little time and practice, but basically it takes two file names or directory specifications as input parameters (so it can have two things to compare to one another). By way of output, it creates a list of all the differences between the two files it finds, using color and other flags to show differences. This is merely handy when comparing process lists, which seldom exceed 40 or 50 items; it's absolutely essential when comparing Windows files or registry values, because they can easily number in the thousands!

That said, how can you tell when a change is significant? As I mentioned in earlier chapters in this book, anything that changes Internet Explorer defaults unexpectedly or unwontedly, adds entries to programs that are run automatically at startup (either in keys that end in \Run, \RunOnce, or buried in class definitions elsewhere in the registry), or removes other entries from those keys (so as to disable firewalls, anti-virus, or anti-spyware software, for example) is suspect. If a little practice doesn't build up your confidence, visit anti-virus and anti-spyware sites and look at the files, registry data, and other items they mention in documenting adware, spyware, and malware, and the items deleted or modified when removing such things manually. These all represent the kinds of things you're looking for and should help you zoom in on your local targets quickly and effectively.

## **Monitoring System Security**

One of the biggest and best improvements in Windows XP SP2 is the introduction of the Security Center. This is a centralized utility that reports on what Windows knows about your system's current level of security, and that provides access to information to address any problems it reports (or at least, advice on what to check to make sure such problems don't exist). On a test PC running Norton Internet Security, for example, although Windows can tell that a firewall and anti-virus software are installed, it apparently can't report on their update status, as shown in Figure 12-11.

You can check in on this utility from time to time to see how Windows thinks you're doing in the security department. On the other hand, each time Windows starts up if there's a need to check status in any of the areas that the Security Center monitors, it'll pop up a warning message that tells you there's something going on that needs looking into. It's a vast step forward over anything else Microsoft has ever done before by way of security monitoring. That said, because it doesn't yet detect all anti-virus programs equally (I tried Norton AntiVirus 2004, BitDefender, and other packages mentioned in Chapter 9, but it could not access status information for all of those it could recognize) nor could it do the same for all third-party firewalls. I imagine this situation will improve as Windows XP SP2 becomes the norm, and more vendors add the necessary hooks into their products to communicate with the Security Center. For example, if I didn't use both the firewall and the anti-virus software built into BitDefender Professional v7.2, the program would report to Security Center that anti-virus software was not enabled, even though it was running and working properly. On the other hand, Norton Internet Security 2005 integrated with Security Center perfectly and would accurately report all status changes in the firewall and anti-virus capabilities separately and correctly. Again, I think this situation should improve with time, as these kinks are worked out where necessary.

Of course, I still think automatic update is the right approach for all security-related software, whether or not Security Center can track its currency and update status. With automatic update turned on and a current subscription, you're guaranteed to be able to keep up with what's likely to show up in your inbox or security perimeter next! My only regret is

that Microsoft didn't choose to include antispam and anti-spyware/anti-adware monitoring features in the Security Center as well. Maybe in Windows XP SP 3?

## Proper Password Handling

I'm going to make some recommendations about password structure and also about how to keep your passwords safe and sound. In an age where many Web sites have passwords, where you probably use a password to log into your Windows computer, and where even some programs and utilities may have passwords, there certainly are enough of them to go around.

So I want to start by shaking the foundations of your universe and say that your password is probably insecure if one or more of the following conditions are true:

- If your password appears in any kind of dictionary, it might be reproduced the same way (or at least from a word list that matches the entries in that dictionary, if not the definitions and other stuff).
- If you use familiar data in or for your password—like the names of your spouse, your children, or your pets, or perhaps your phone number, street number, or part of your Social Security number—crackers often customize their dictionaries with such data when attacking you.
- Same goes for birthdays, anniversaries, and other numbers that relate to you and your loved ones.

Since I just described most passwords that people use, what's a person to do? The answer lies in a good working understanding of password complexity. A sufficiently complex password is much more difficult to guess, and makes whatever that password protects much less likely to succumb to a dictionary-based attack. But what are the ingredients of a complex password? Glad you asked! According to Microsoft, and lots of other experts who provide password guidelines, a complex password is or contains:

- At least 8 characters, preferably as many as 14
- A mix of upper- and lowercase letters, numbers, punctuation marks, and other symbols
- Is sufficiently strange and random to be difficult to guess, and unlikely to be in anybody's dictionary
- Follows some logic you understand, or some structure you can re-create, but that's unlikely for somebody else to be able to do likewise (unless you tell them, in which case you've violated a major password security rule)

Most dictionary attacks are smart enough to try obvious substitutions for vowels (@ for a, 3 for e, 1 for i, 0 for o, and so forth) so please don't fall prey to the idea that simple replacements for dictionary words gets you off the hook, either. An old friend and colleague of mine likes to explain what this means by using the example password `le4PoTw/3l:Ps&O` as an acronym for "I eat four pizzas on Thursdays, with three ingredients: pepperoni, sausage, and onions." Note that every other alphabetic character is upper- or lowercase, and there are a couple of numbers and three punctuation marks for good measure (and good complexity) thrown in. Use this approach as an example, but don't use this password, please: because it's in print, it just might show up in somebody's dictionary for that reason!

Next, here are five simple rules for passwords that you should violate only at your peril:

- Never write down passwords, unless they're stored in a very secure location (preferably a safe, but hidden in a locked drawer or lockbox is okay).
- Don't share your passwords with anybody. You never know when they'll violate any of the password rules. Administrators, bosses, and security staff are the only possible exceptions.
- Never e-mail your password to anybody (besides, doing so violates the previous rule, right?).

- Change your passwords regularly—at least every 6 months or so (frequency usually varies by how sensitive the materials and information you work with might be: in government top-secret workplaces, they routinely change passwords monthly, and sometimes, even more often than that).
- Don't use the same password for multiple sites, logins, or other password-protected assets. Otherwise, compromising one can lead to compromising them all (or as many as share the same password, anyway).

"Holy cow!" I hear you saying, "I need about 20 passwords! How am I going to remember all that stuff?" Good question! Fortunately, oodles and scads of password manager programs are available nowadays, so the only password you really need to remember is the one that unlocks that program (but that means it better be a really good password, comprende?). Numerous commercial password managers are available, but I mention a handful of my favorite freeware tools here believing that buying some or all of a firewall, anti-virus, anti-spyware/anti-adware, and possibly even antispam software or services has probably depleted your budget somewhat by now. See Table 12-2 for some recommendations (use your favorite search engine with "free password manager" as a search string if you decide you don't like any of these).

Table 12-2 A Handful of Free Password Manager Programs		
Name	Description	URL
RoboForm	Password generation, storage, and autotext app	<a href="http://www.roboform.com/">www.roboform.com/</a>
HyperSafe	Provides local or Web-based access to passwords	<a href="http://www.passwordsafe.com/">www.passwordsafe.com/</a>
KeyWallet	Provides local password storage and access	<a href="http://www.keywallet.com/">www.keywallet.com/</a>
Password Safe	Bruce Schneier's open source password safe	<a href="http://www.schneier.com/passsafe.html">www.schneier.com/passsafe.html</a>
Secure Data Manager	Open source password manager with annotations	<a href="http://sdm.sourceforge.net/">http://sdm.sourceforge.net//td&gt;</a>

Grab one and use it with your newly invigorated and incredibly innovative collection of passwords. For my own part, I'm entranced with Schneier's Password Safe (he's a real star in the computer security world, and his stuff is great) as well as the Secure Data Manager (also known as SDM). Other possible do-it-yourself approaches might include creating password-protected files in Word or Excel, or perhaps using a password manager built into a third-party browser (Internet Explorer will happily manage passwords for you, too, but its protection schemes have been cracked enough in the past for me to be nervous about recommending that approach without this warning).

### Stay Away from Risky Downloads

It's a truism I've mentioned throughout this book that most unwanted content and software arrives by invitation on most PCs, rather than by insidious or nefarious means. At this point, I assume you're convinced that threats are everywhere and that vulnerabilities can be exploited given the right opportunity. If you've installed a firewall, anti-virus software, anti-spyware/anti-adware software and have done what you can to protect your system from these threats, that doesn't mean you can do anything you want on the Internet.

It's important to recall that signatures and other means of positive identification inform most of what protective software can do for your PC. Indeed, the presence of anti-virus and anti-spyware/anti-adware software on your machine should protect you from known threats—but what about new or unknown ones? I look at software downloads much the same way as I do at e-mail attachments: okay if they come from a known and trusted source, but questionable if not downright dangerous otherwise.

To make my point as directly as possible, don't download software from unknown or untrusted sources. If you can't find a glowing review of some shareware or freeware program in a reputable publication or on a well-known Web site, you're tempting fate (and risking infection or infestation) if you copy a download to your PC, and then install the software it contains. Stick to well-known sources of shareware and freeware and resist the temptation to grab a cool-sounding or -looking tool or utility. Just because you can download anything you want, doesn't necessarily mean that you should.

### When in Doubt, Play It Safe!

When you're working with your PC, cruising the Internet, reading e-mail, or diverting yourself in some hopefully enjoyable way, don't take unnecessary chances with unknown and potentially unsafe materials. Even though there is often some subterfuge or covert activity involved when unwanted software makes itself at home on a PC, it usually enters that machine through the front door, buried inside some supposed prize or possible treasure that users download. Although

the protective software you install on your PC should protect you from routine threats, it's just not smart to open the door to potential infestation or infection.

The key to playing it safe is to do some homework before downloading anything. The best way into a download is through a link provided in a reputable publication (such as PC Magazine and other well-known publications that cover computing topics, tools, and technologies) or from a Web site that you know and trust (elsewhere in this book, I've cited sites such as The Ultimate Collection of Windows Software a.k.a. [tucows.com](http://tucows.com), CNET's [Shareware.com](http://Shareware.com), ZDNet at [www.zdnet.com/downloads](http://www.zdnet.com/downloads), and so forth). Even if you find pointers to a program somewhere else on the Web, if the program's got sufficient capability and has generated real interest in the user community, you can probably find a copy of somewhere safer—if you take the time to look. Save yourself some possible grief, and do just that! Resources

Legions of great resources are available that explain what processes run on a Windows machines, which ones are benign and necessary, which ones are benign and possibly unnecessary (and how to do away with them if you decide you don't need them), and which ones are potentially dangerous or outright malign. I found three stellar resources while researching this chapter, but given the time I know I could find more.

- There's very good built-in process info at the "I am Not a Geek" (sez you!) Web site at [www.iamnotageek.com/](http://www.iamnotageek.com/), but a search engine like Google seems to be the best way to dig into its contents because I couldn't find many of the articles Google turned up for me by trying to navigate my way into that site top-down. If you simply search on process names, you'll find this site popping up repeatedly, so why not just take the most obvious approach?
- The Los Angeles Free Net has a great collection of Web pages called "Startup Programs and Executables Listing" that includes links to information for a sizable and reasonably comprehensive collection of process image names ([www.lafn.org/webconnect/mentor/startup/PENINDEX.HTM](http://www.lafn.org/webconnect/mentor/startup/PENINDEX.HTM)).
- Paul Collins maintains a decidedly comprehensive startup programs list that includes most process executables plus a raft of other items; you can access a search engine against that list at [www.sysinfo.org](http://www.sysinfo.org) or jump straight to the list at [www.sysinfo.org/startuplist.php](http://www.sysinfo.org/startuplist.php).

If you're going to dig into the many command-line utilities that Windows supports, please avail yourself of Windows XP's built-in "Command Line Reference" for syntax information and examples to help you get things right. To access this reference, choose Start@ @>Help and Support, and then type command line reference into the Search box in the upper left-hand corner of the resulting screen.

You can learn more about the WinTasks program at [www.liutilities.com](http://www.liutilities.com). I also found ready access to some, but not all, of the process information from my Task Manager list on its site by typing URLs constructed as follows: <http://www.liutilities.com/products/wintaskspro/processlibrary/<img>/>, where you substitute the image name without the .exe extension for <img> (so that looking up the Application Layer Gateway service, or alg.exe, would use /alg/ at the end of the aforementioned URL).

Although I already mentioned Jerry Honeycutt's outstanding book Microsoft Windows XP Registry Guide (Microsoft Press, 2002) in Chapter 4, because he explains how to compare registry versions using WinDiff therein, I think it's worth another mention here. He also wrote a peachy article for Microsoft entitled "[Safekeeping the Windows XP Registry](#)."

Microsoft's WinDiff utility is an amazing tool, if you're willing to take the time to learn how to use it. To that end, you'll find Microsoft Knowledge Base Article 159214 "How to Use the Windiff.exe Utility" extraordinarily informative (<http://support.microsoft.com/default.aspx?scid=kb;en-us;159214>). If you're not really interested in lots of cryptic character display Chris Maunder has created WinDiff UI, a graphical interface for the program that's much easier to use and understand than WinDiff itself. You can read more about this tool and download an executable at [www.codeproject.com/tools/runwindiff.asp](http://www.codeproject.com/tools/runwindiff.asp).

The Windows XP product documentation includes a short but detailed technical description entitled "[Password must meet complexity requirements](#)". To see the company's official take on what makes a password sufficiently complex, please consult that Web page.

## Summary

This chapter offered numerous tips and techniques for practicing system safety. In particular I explored the process of creating a system baseline to use as a comparison if things ever start to get weird on your computer, as well as some thoughts regarding monitoring system security, managing passwords, and some commonsense rules for downloading anything off the Internet.

The next chapter moves on to the final part of this book, and changes focus to reviewing the kinds of regular security routines that you should practice. I explore a regular security regimen in Chapter 13 to help you keep up with the current state of security (whatever it may be), and also describe in Chapter 14 the kind of automated scans and checks you should be performing on your PC on a regular basis. The idea is to maintain a level of security awareness and checks that will minimize the chances of an unpleasant surprise appearing from out of the blue!