

Realtime
publishers

"Leading the Conversation"

The Definitive Guide™ To

Information Theft Prevention

sponsored by

Blue  **Coat**[®]

Dan Sullivan

Chapter 5: Protecting Information Use on Unmanaged Devices.....	87
Threats to Information Delivered to Unmanaged Devices	88
Keyloggers and Information Loss.....	89
Structure and Function of Keyloggers	89
Hook-Based Keyloggers	90
Driver Keyloggers.....	93
Hardware-Based Keyloggers	94
Video Buffer Capture Programs	94
Browser Buffer Cache Vulnerabilities.....	95
Caching Basics.....	96
Spyware and Information Leaks	98
Clipboard Vulnerabilities.....	98
Browser Cookie Vulnerabilities.....	98
Browser Helper Object Threats	98
Creating Secure Zones within Unmanaged Devices.....	101
On-Demand Security Measures	101
On-Demand Security Programs and Unmanaged Devices	102
Protecting Information on Unmanaged Devices.....	104
Assessing Security Profiles.....	105
Disabling Features based on Security Profile	106
Terminating Sessions	108
Summary	108

Copyright Statement

© 2006 Realtimepublishers.com, Inc. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtimepublishers.com, Inc. (the “Materials”) and this site and any such Materials are protected by international copyright and trademark laws.

THE MATERIALS ARE PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice and do not represent a commitment on the part of Realtimepublishers.com, Inc or its web site sponsors. In no event shall Realtimepublishers.com, Inc. or its web site sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

The Materials (including but not limited to the text, images, audio, and/or video) may not be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, in whole or in part, except that one copy may be downloaded for your personal, non-commercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

The Materials may contain trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties.

Realtimepublishers.com and the Realtimepublishers logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners.

If you have any questions about these terms, or if you would like information about licensing materials from Realtimepublishers.com, please contact us via e-mail at info@realtimepublishers.com.

[**Editor's Note:** This eBook was downloaded from Realtime Nexus—The Digital Library. All leading technology guides from Realtimepublishers can be found at <http://nexus.realtimepublishers.com>.]

Chapter 5: Protecting Information Use on Unmanaged Devices

The widespread adoption of the Internet has changed many aspects of information service delivery. When mainframes were the predominate computing resources, users would employ terminals directly connected to the computer. The advent of mini-computers in the 1970s and even the early adoption of personal computing did not change the model; serial connections from a terminal or the PC was the most common means of communication between the computing service and the client device. This method was a highly restrictive means of communications. Access to information resources was limited to those who could work in a computing center or a building wired to the computing center. The precursors of the Internet were the beginning of what would become a radical shift in computing communications.

Research into networking began in the early 1960s, and the first wide area network (WAN), albeit slow and primitive, was built in 1965. Work on the ARPANET, the predecessor of the Internet, began in 1968. By the late 1990s, the Internet was widely deployed and access to computing resources was no longer limited to direct connection terminals and PCs. This shift introduced another new concept to systems administration, the unmanaged device.

Unmanaged devices are computers, personal digital assistants, cell phones, and other devices that can access an organization's network but are not necessarily controlled by that organization. Consider a typical bank. In the past, the only way to get information about the status of your accounts was to walk into a branch office and speak to a teller who had a dedicated terminal connected to the mainframe that ran the bank's software. Advances in telecommunication technologies changed this model with the advent of telephone access and automated teller machines. The bank, however, still had strict control. In the case of telephones, the communication mechanism is relatively simple and not as vulnerable as computer networks. ATMs, of course, are the managed devices controlled by the bank.



The telecommunication system had its own version of hacking, known as "phreaking." The object was to send control signals through telephone lines to essentially trick the system into allowing for unpaid services, such as long distance calling.

Banking information is now readily available over the Web. You do not have to go any further than your PC to check balances, pay bills online, or transfer funds between accounts. This is a great advance for many of us; it's a challenge for systems administrators.

The problem with introducing unmanaged devices is the damage that can be done with them, either intentionally or not. Consider the following scenarios:

- A home user has a PC infected with a variety of malware that spreads when the PC is accessing the corporate network.
- A road warrior accesses the corporate network from a PC in a hotel business center and leaves confidential documents cached on its local hard drive.
- A corporate laptop is infected with a Trojan horse while connected to a home network and is now a member of a botnet used for emailing spam, even when connected to the corporate network.
- A bank customer uses a mobile device and a wireless network in an airport to check account balances while waiting for her plane. Her username and password are stolen by hackers scanning the wireless network.

In each case, information is compromised without malice or incompetence. Average users working with their devices in average ways are vulnerable to information loss. As this chapter explores, traditional security mechanisms that depend on being able to manage all devices connected to an organization's IT infrastructure are no longer sufficient.

Threats to Information Delivered to Unmanaged Devices

When information leaves the controlled IT infrastructure of an organization's IT infrastructure, it is subject to several threats:

- Keyloggers
- Video buffer capture programs
- Theft of data from browser caches

Each of these threats demonstrates the need to protect information where it goes, not just at its source.

Keyloggers and Information Loss

Keyloggers are programs that record keystrokes as they are typed on a keyboard. This content sounds pretty mundane, and in most cases it is. Very few information thieves would be interested in the contents of a typical user's email messages, memos to colleagues, or updates to project plans. Usernames and passwords, however, are a different story. Imagine a stream of text like the following being captured by a keylogging program:

```
www.google.com
football scores
new england patriots
jets
www.midatlanticbankandtrust.com
johndoe
harley27
0014778718938
news.yahoo.com
...
```

It's clear that the text is being typed into a Web browser. Most of the steam of text is the product of browsing for sports scores and other news. The interesting content begins with the Web address of a (fictitious) bank, www.midatlanticbankandtrust.com. Chances are good that the text immediately following the URL of the bank is a username, password, and bank account number. Like panning for gold, sifting through large volumes of useless data can sometimes yield a small amount of very valuable information. This kind of information is, unfortunately, relatively easy to capture.

Structure and Function of Keyloggers

Keylogging may sound difficult at first. How could one key logging program possibly know all the different ways other programs will prompt for text or take input? After all, Microsoft Word is different from Mozilla Firefox, which is different from a custom data entry form used with a database application. The trick is to focus not on the application level but on the operating system (OS) level.



For the purpose of this discussion, the focus will be on the Windows family of operating systems (OSs). The model presented here might match the methods used by other OSs to varying degrees, but there will be differences as well.

Keyloggers, in general, use three methods to capture keystrokes:

- Intercept OS messages passed from the keyboard driver to the OS
- Employ driver keyloggers that operate at trusted layers of the OS
- Use hardware devices to intercept keystrokes

Intercepting OS messages is the most commonly used method. It takes advantage of an OS feature known as hooks.

Hook-Based Keyloggers

OSs, such as those in the Windows family, use a message passing mechanism to control the proper processing of events within the multitasking OS—events such as keystrokes, mouse clicks, mouse movements, window drawing, and other changes in the environment that requires the OS to respond. Modern OSs are busy. They support multiple processes running within the system as well as multiple events occurring nearly simultaneously. Something needs to keep these events and applications coordinated. For example, you can have a Web browser, a spreadsheet, and a multimedia application running at the same time without having to worry that data typed into the browser will end up in the spreadsheet or the data from a CD will end up anywhere other than with the media player. The “traffic cop” within the Windows OS is the message queue and message dispatcher (see Figure 5.1).

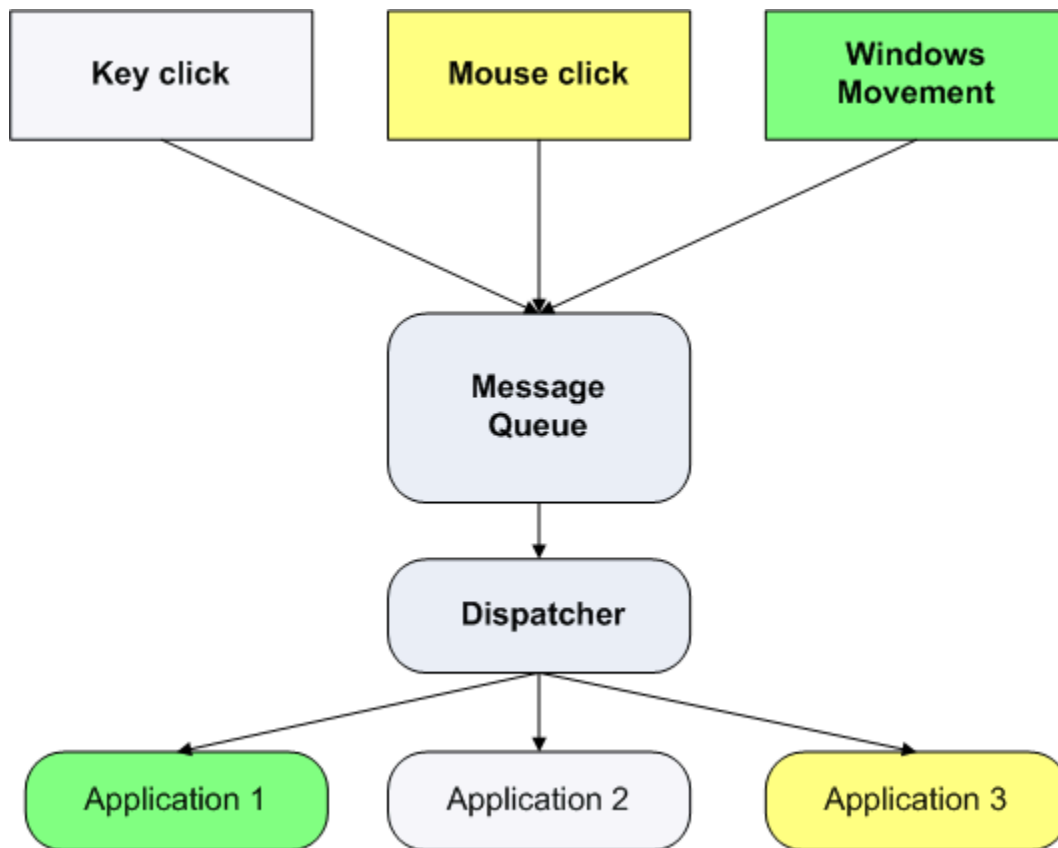



Figure 5.1: The OS message queue ensures all events are captured and held until the appropriate application can process the event.

 The technical details of how messages are represented and dispatched are beyond the scope of this chapter. For details about this process, see “Understanding the Message Loop” at http://www.winprog.org/tutorial/message_loop.html.

There are clear advantages to this model compared with the setup of earlier OSs, such as DOS. With a message queue and dispatch mechanism, you are not limited to running one program at a time. You also have a way to monitor what is going on inside the OS by watching the message queue.

This ability can be useful when debugging software. For example, when testing an application, a monitor can record all the events in a message queue and when an error occurs, developers will have detailed information about the context of the error. The trick is to monitor the message queue without interfering with its normal operations. The monitoring program, for example, should not take a message out of the queue the way the dispatcher does; it should just copy information about the message and leave it for processing by the appropriate application.

This setup is accomplished with a mechanism known as a hook. Hooks are functions that are called by the OS when a specific event occurs. Hooks can examine messages for only one application (known as local hooks) or for the entire system (global hooks). There can be multiple hooks for the same event, in which case, one hook is called after another (see Figure 5.2).

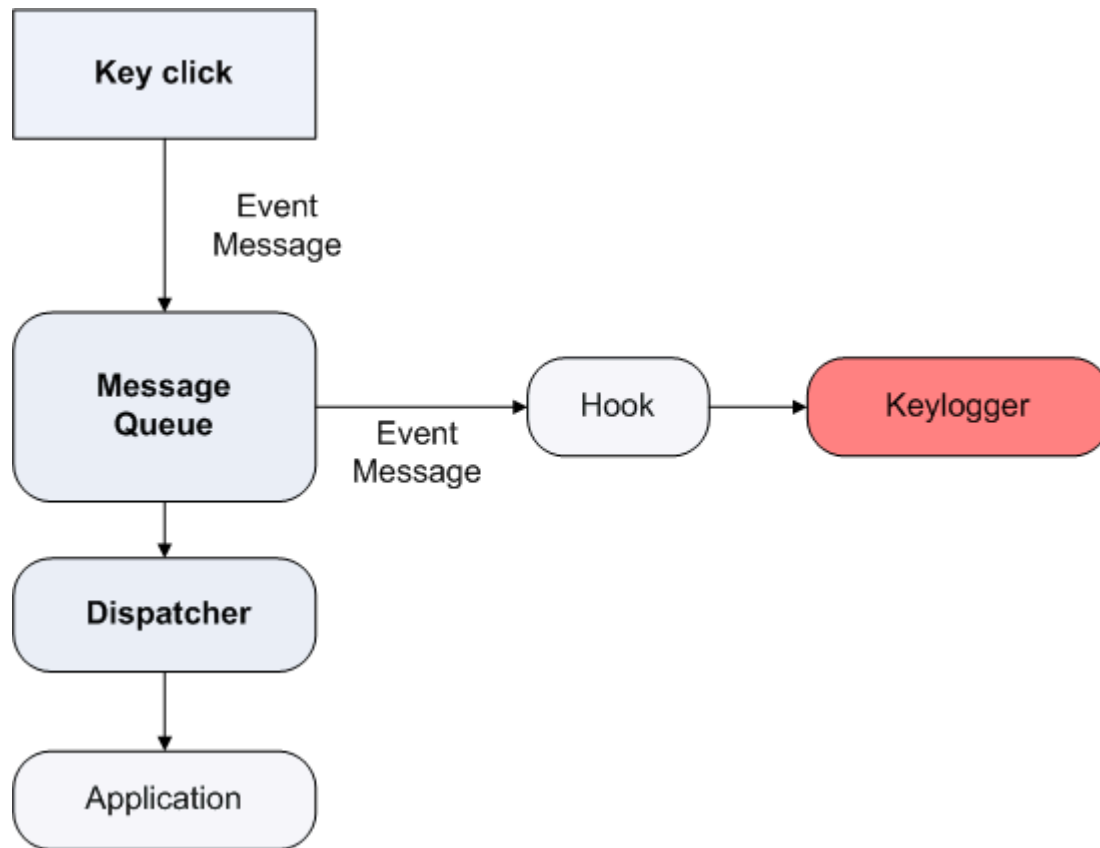


Figure 5.2: Hooks can be used to capture information for use by keyloggers, spyware, and other malicious programs.

Regardless of their sophistication with regards to taking advantage of OS features, software keyloggers can be detected and blocked by on-demand security mechanisms, as Figure 5.3 shows.

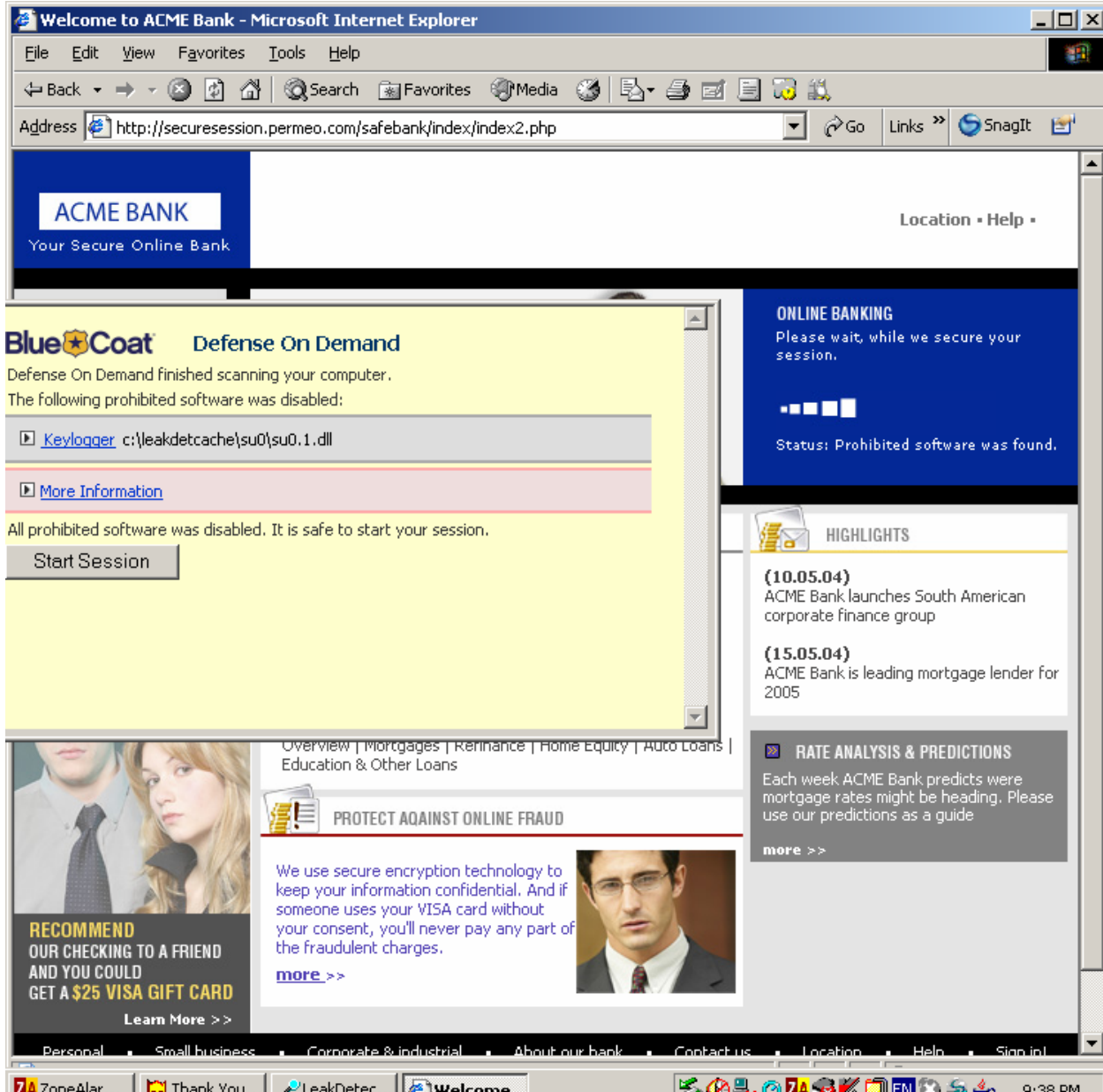


Figure 5.3: Keyloggers and other spyware programs can be detected and terminated by on-demand security mechanisms.

Driver Keyloggers

Driver keyloggers are able to bypass the message queue entirely by receiving keyboard data directly from the keyboard. These loggers are installed in the OS and replace the legitimate driver designed to receive data from the keyboard. These programs run in the most secure level of the OS (kernel or executive mode) and are therefore difficult to detect with application-level anti-spyware programs. Figure 5.4 shows a basic version of the Windows OS modes and the processes that run in each.

For more information about the Windows OS modes, see Steven Roman's "Windows Architecture" at <http://www.microsoft.com/technet/archive/ntwrkstn/evaluate/featfunc/winarch.msp?mfr=true>.

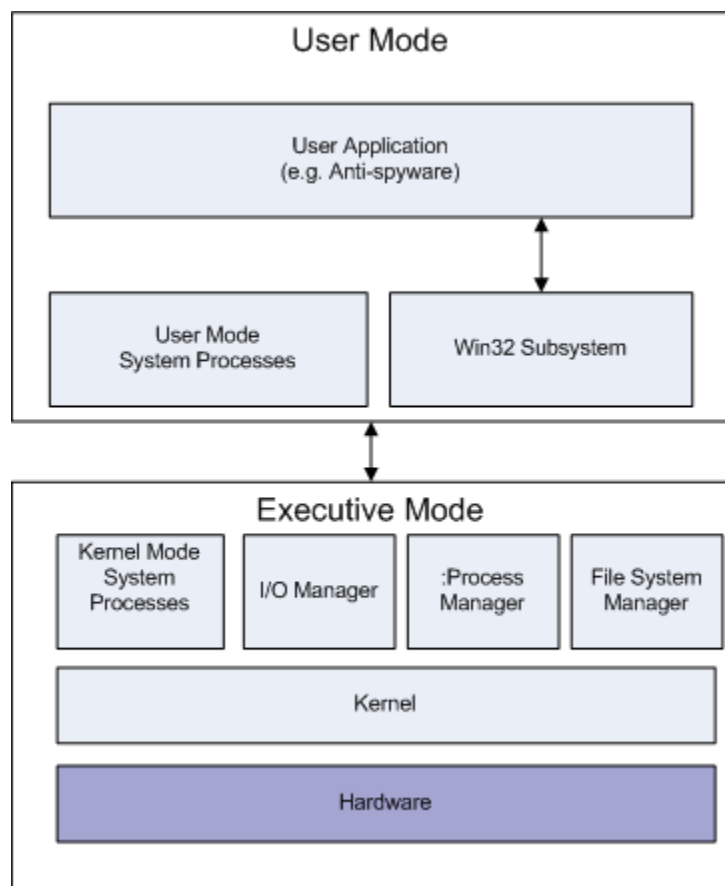


Figure 5.4: The Windows architecture separates user mode and executive mode programs; detecting and stopping executive-mode programs using user-mode tools is more challenging than detecting and stopping user-mode programs.

In addition to monitoring the message queue, keyloggers can be hardware based.

Hardware-Based Keyloggers

Hardware keyloggers are devices that are placed between the keyboard and the keyboard plug on the computer. These are physical devices, so they are not useful for spyware and typically have legitimate uses:


- Investigating suspected violations of organizational policy or laws
- Auditing activity on secure devices
- Software testing and debugging
- Training and evaluation

From a security perspective, hardware keyloggers are less of a threat than software-based keyloggers. Keyloggers are a threat to information security because they can record text typed by a user without the user's knowledge. The collected information is then emailed or transferred by ftp or similar method to the attacker. Depending on the sophistication of the keylogger, it might perform preprocessing on the collected information. For example, it might filter the text so that only text around URLs (the most likely location of usernames and passwords) is sent.

Keyloggers have their limitations, at least from the perspective of information thieves. Keyloggers capture only one side of a communication, such as when a user types an email but not when the user reads one. Similarly, keyloggers do not capture information from earlier stages of an incremental development; for example, when modifying an existing spreadsheet, the keylogger captures only the changes, not the existing content. Needless to say, hackers have found a solution to these limitations.

Video Buffer Capture Programs

Video buffer capture programs are malicious programs that make copies of the video buffer within a computer at periodic intervals. Like keyloggers, these programs take advantage of the internal structure of the OS and hardware mechanisms that provide basic I/O services.

 Video buffer capture programs should not be confused with two legitimate types of programs with similar names. Screen capture programs, such as SnagIt (<http://www.snagit.com/>), are utilities for copying parts of a screen to an image, which can be saved as a file or embedded in a document. Video capture programs copy frames from video capture devices, such as Webcams. Again, these are end user programs, not malicious programs installed and operated without the knowledge of the user. Figure 5.5 shows LeakDetector, a free vulnerability assessment tool offered by Blue Coat that simulates various types of spyware, including a buffer capture program.

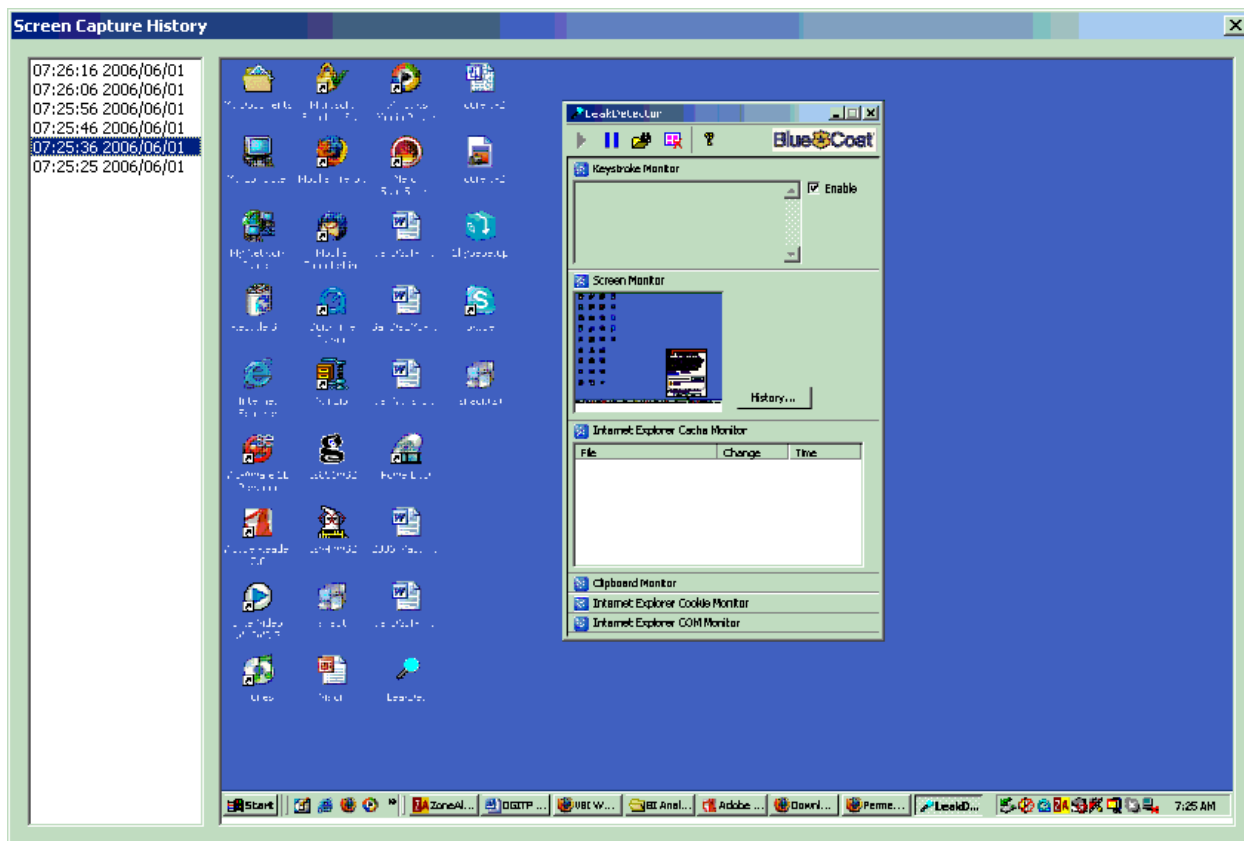


Figure 5.5: Blue Coat LeakDetector simulates a video buffer capture program that collects screenshots without a user's knowledge.

Consider this scenario: A salesperson is attending a conference and uses the hotel business center to check email. One of the emails is from the salesperson's manager and includes an attached Excel spreadsheet with sales forecasts. The user opens the spreadsheet, updates some numbers, and sends the spreadsheet back. Now, if only a keylogger were enabled on the computer, a series of numbers, without context, would be sent to the attacker that infected the computer. With a video grabber, however, the full context can be seen. Video grabbers are one method for capturing the broader context of a user's session; another is capturing data from browser buffer caches.

Browser Buffer Cache Vulnerabilities

Browsers have become ubiquitous tools for computer users, attaining the same status as a "must have" program as word processors and email clients. Along with their increasing popularity, browsers have improved to provide more functionality and better performance. For a long time (at least relative to the history of browsers), Web browsers have used caching as a means to improve performance.

Caching Basics

The idea behind caching is simple: data that has been recently used is likely to be used again in the near future. Caches are used in applications and hardware. In fact, processor manufacturers often include specifications for their level-1 and level-2 CPU caches. Large, enterprise-scale databases use caches to keep data in memory rather than having to return to the disk to retrieve data again. Database designers opt to implement data caches even though it makes the application more complex because retrieving data from memory can be hundreds of times faster than retrieving from disk. The same argument applies to Web browsers, except the difference in time required to retrieve data over the Internet is far greater than the time needed to retrieve data from a local disk.

In the case of browsers, the data that is saved is previously retrieved pages. This makes some navigation quite fast. Clicking the back arrow on a browser typically causes a page to be retrieved from cache instead of from a Web server.

As Figure 5.6 shows, users can change the size of the cache, which limits the amount of data that can be stored in the cache. They can also clear the cache at any time—not that many do. Herein lies the problem—when the cache is not cleared on a secured, managed device, it can be compromised by malware or by someone with physical access to the machine and logon credentials. In the case of public use machines, that someone with physical access can be anyone.

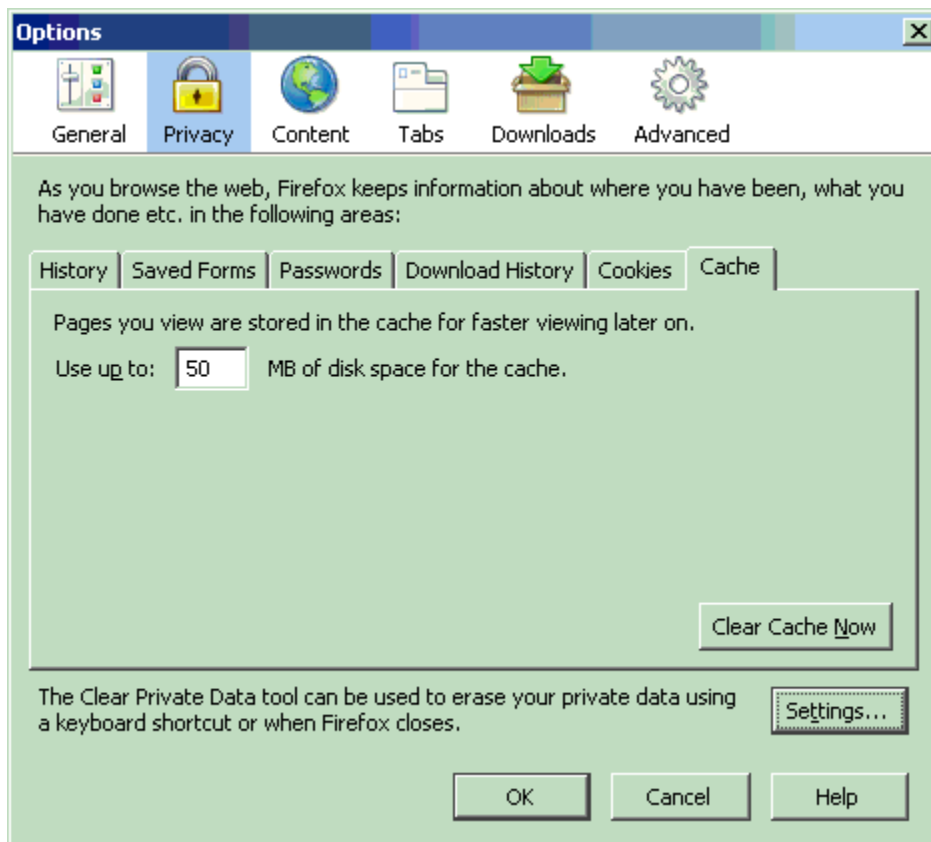
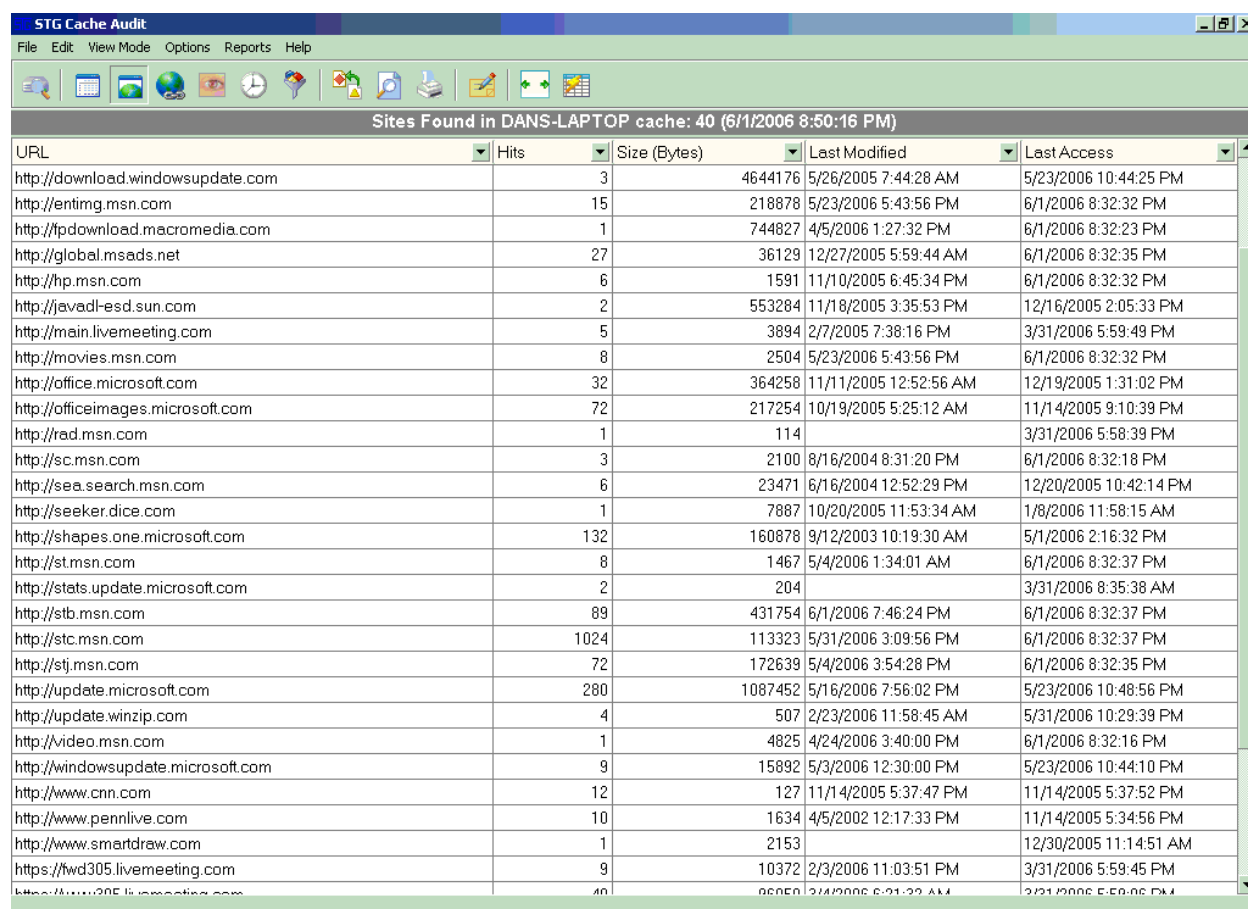


Figure 5.6: The size of browser caches can be controlled by users, who can also delete the contents of the cache at any time.

Imagine you are at an industry tradeshow. For convenience, conference organizers set up laptops for general use in the exhibit hall. Salespersons, marketing executives, product managers, and other industry professionals use the laptops to check emails, including reviewing attachments through the Web. Would you like your competitors to know what you are working on? Who your competitors are soliciting? The data you just looked up on SalesForce.com? If not, you would be advised to clear the browser cache before leaving the device.

In some cases, it might be just as easy as clicking the back button to find out what has been stored in the cache. For the more sophisticated users, freely available utilities can be used to analyze and query the contents of the browser cache. As is too often the case, tools such as browser cache utilities designed for forensic and audit purposes can be used for information theft as well (see Figure 5.7, which shows a screenshot taken from STG Cache Audit available at <http://www.stgsys.com/audit.asp>).



URL	Hits	Size (Bytes)	Last Modified	Last Access
http://download.windowsupdate.com	3	4644176	5/26/2005 7:44:28 AM	5/23/2006 10:44:25 PM
http://entimg.msn.com	15	218878	5/23/2006 5:43:56 PM	6/1/2006 8:32:32 PM
http://fpdownload.macromedia.com	1	744827	4/5/2006 1:27:32 PM	6/1/2006 8:32:23 PM
http://global.msads.net	27	36129	12/27/2005 5:59:44 AM	6/1/2006 8:32:35 PM
http://hp.msn.com	6	1591	11/10/2005 6:45:34 PM	6/1/2006 8:32:32 PM
http://javadl-esd.sun.com	2	553284	11/18/2005 3:35:53 PM	12/16/2005 2:05:33 PM
http://main.livemeeting.com	5	3894	2/7/2005 7:38:16 PM	3/31/2006 5:59:49 PM
http://movies.msn.com	8	2504	5/23/2006 5:43:56 PM	6/1/2006 8:32:32 PM
http://office.microsoft.com	32	364258	11/11/2005 12:52:56 AM	12/19/2005 1:31:02 PM
http://officeimages.microsoft.com	72	217254	10/19/2005 5:25:12 AM	11/14/2005 9:10:39 PM
http://rad.msn.com	1	114		3/31/2006 5:58:39 PM
http://sc.msn.com	3	2100	8/16/2004 8:31:20 PM	6/1/2006 8:32:18 PM
http://sea.search.msn.com	6	23471	6/16/2004 12:52:29 PM	12/20/2005 10:42:14 PM
http://seeker.dice.com	1	7887	10/20/2005 11:53:34 AM	1/8/2006 11:58:15 AM
http://shapes.one.microsoft.com	132	160878	9/12/2003 10:19:30 AM	5/1/2006 2:16:32 PM
http://st.msn.com	8	1467	5/4/2006 1:34:01 AM	6/1/2006 8:32:37 PM
http://stats.update.microsoft.com	2	204		3/31/2006 8:35:38 AM
http://stb.msn.com	89	431754	6/1/2006 7:46:24 PM	6/1/2006 8:32:37 PM
http://stc.msn.com	1024	113323	5/31/2006 3:09:56 PM	6/1/2006 8:32:37 PM
http://stj.msn.com	72	172639	5/4/2006 3:54:28 PM	6/1/2006 8:32:35 PM
http://update.microsoft.com	280	1087452	5/16/2006 7:56:02 PM	5/23/2006 10:48:56 PM
http://update.winzip.com	4	507	2/23/2006 11:58:45 AM	5/31/2006 10:29:39 PM
http://video.msn.com	1	4825	4/24/2006 3:40:00 PM	6/1/2006 8:32:16 PM
http://windowsupdate.microsoft.com	9	15892	5/3/2006 12:30:00 PM	5/23/2006 10:44:10 PM
http://www.cnn.com	12	127	11/14/2005 5:37:47 PM	11/14/2005 5:37:52 PM
http://www.pennlive.com	10	1634	4/5/2002 12:17:33 PM	11/14/2005 5:34:56 PM
http://www.smartdraw.com	1	2153		12/30/2005 11:14:51 AM
https://wd305.livemeeting.com	9	10372	2/3/2006 11:03:51 PM	3/31/2006 5:59:45 PM
https://www.205.livemeeting.com	40	96950	2/4/2006 6:31:32 AM	3/31/2006 5:59:46 PM

Figure 5.7: Freely available tools can be used to query the contents of the browser cache.

So far, this chapter has discussed individual threats to information from unmanaged devices. Keyloggers, video frame grabbers, and browser cache utilities can all be used to collect information from unmanaged devices. Let's consider an example that examines these and related vulnerabilities together.

Spyware and Information Leaks

Once a piece of spyware has infected a device, it can capture information in several ways:

- Keylogging
- Video frame grabbing
- Monitoring browser caches
- Copying contents of the Windows clipboard
- Monitoring browser cookies
- Monitoring browser COM objects

Clipboard Vulnerabilities

We often do not think much of the Windows clipboard except when we need it. Want to copy a table of budget figures from an Excel spreadsheet to Word? Almost without thinking, we hit Ctrl+C, navigate to the other application, hit Ctrl+V, and we are done. Unfortunately, because of the flexibility of the Windows OS, the contents of the clipboard are available to other applications as well, including spyware.

Browser Cookie Vulnerabilities

Browser cookies alone are not necessarily all that useful to information thieves, but when combined with other pieces of information—such as an account number or username—they can become quite valuable. Again, spyware can easily access cookies, which are simply stored in files on PCs.

Browser Helper Object Threats

COM objects are programming components used in the Windows environment. One class of these, Browser Helper Objects (BHOs), is designed to give developers control over Microsoft Internet Explorer (IE). BHOs have been the basis for useful programs, such as robust file downloaders, and have been used to develop helpful toolbars and other applications. Spyware writers have used BHOs as another means to capture information and to gain some degree of control over user's computers, as Figure 5.8 shows. (This figure shows a screenshot for the BHO detection tool BHODemon from Definitive Solutions, <http://www.definitivesolutions.com/>. BHODemon is a free utility.)

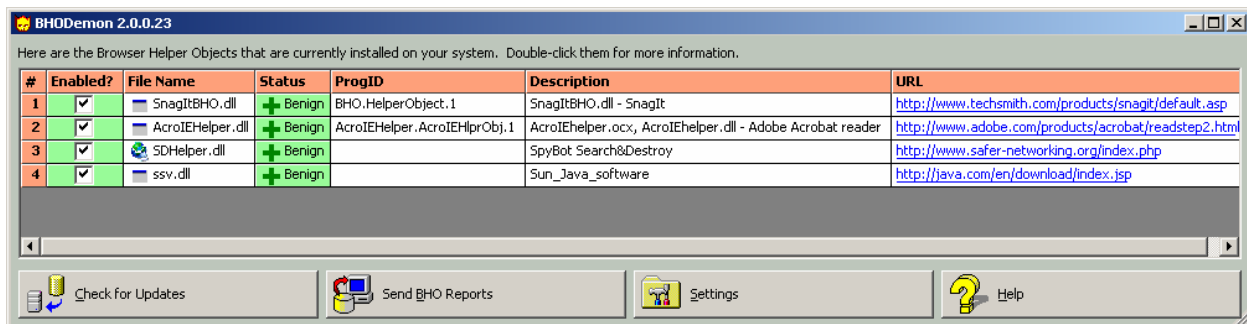


Figure 5.8: BHOs can be either benign or malicious. Utilities such as BHODemon can identify installed BHOs and indicate their benign/malicious status.

Unmanaged devices are subject to several threats and vulnerabilities that can compromise the integrity of distributed, Web-based systems. Furthermore, it is often the case that multiple threats operate in concert to collect confidential information. Using LeakDetector, a vulnerability scanner that displays information accessible to spyware and other information gathering malware, you can readily see the kinds of information that is exposed through conventional browser technology.


 LeakDetector is a free browser vulnerability scanner available from BlueCoat at <http://www.permeo.com/DoD/beta/>.

Figure 5.9 shows the results of LeakDetector monitoring a typical browser session. In this example, a user visits an example bank site for routine banking tasks, such as ordering checks and reviewing account balances. In a matter of minutes, the following information became vulnerable to capture by spyware:

- Username and password
- List of account transactions
- A list of Web site URLs that happen to be on the clipboard
- Information about BHOs currently in use

Although the safeguards deployed by the bank to protect its information internally might work quite well and the bank's efforts to prevent data leaks during transmission by encrypting data are equally effective, once the data arrives at an unmanaged device and is unencrypted, it becomes vulnerable.

Creating Secure Zones within Unmanaged Devices

Problems encountered with information security are analogous to security problems confronted in the physical world. Consider, for example, the security around a head of state or other major political figure. Heads of state are well protected in their own residences and offices. The White House uses multiple security measures and monitoring devices to protect the President of the United States. Similarly, corporate networks are well protected with established and carefully deployed countermeasures to network threats.

When a head of state travels, he or she is surrounded by bodyguards, driven in bulletproof vehicles, and flown in highly secure airplanes. The head of state becomes effectively inaccessible to outsiders. Encrypted data is similarly inaccessible.

Of course, no one, not even heads of state, can stay out of the public constantly. There are times when they have to move about in less-controlled environments, analogous to data moving to unmanaged devices. In the physical world, the less-controlled environment is first prepared by an advance team of security experts that assess the situation, determines appropriate controls, protects their charge, and once he or she is done and leaves the area, the advance team leaves as well. A similar process is needed for protecting data on unmanaged devices.

On-Demand Security Measures

On-demand security measures are policies and programs that are used when data that should be protected is sent to unmanaged devices. Policies define what actions are allowed under particular conditions; the on-demand security programs enforce those policies. The steps for deploying on-demand security are:

- Downloading security programs to unmanaged devices
- Assessing the security profile of unmanaged devices
- Controlling operations on the unmanaged device according to the security policies
- Removing the on-demand security program when the session is terminated

The on-demand security measures serve analogous purposes to advance teams dealing with physical security.

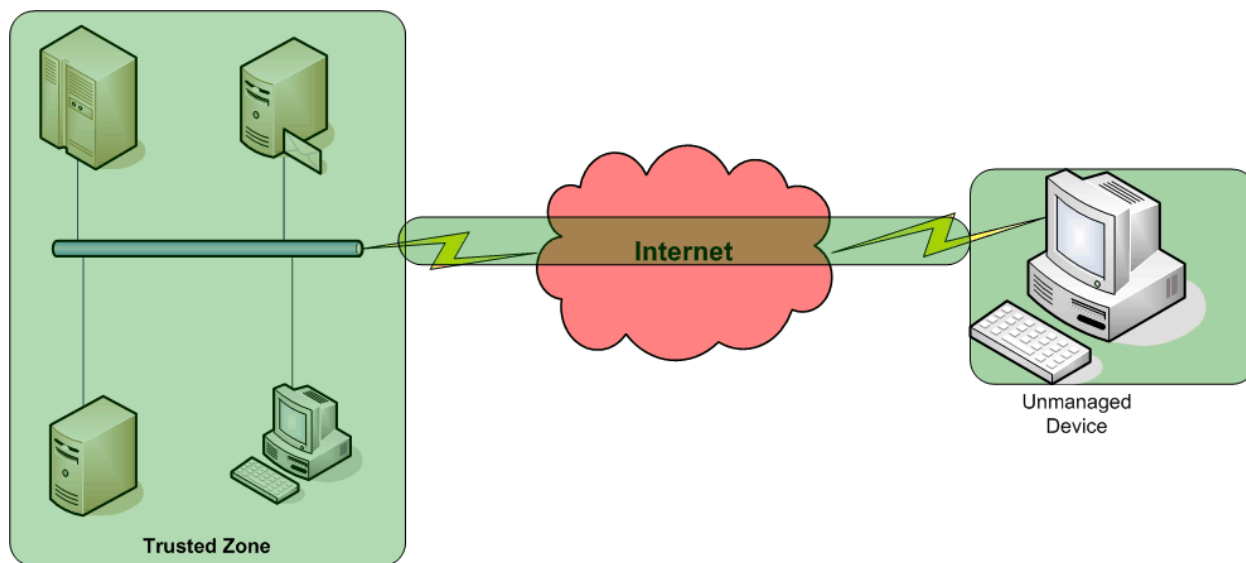


Figure 5.10: Security on demand is about extending the zone of trust that exists within managed device environments.

On-Demand Security Programs and Unmanaged Devices

On-demand security must work within a larger set of constraints than typical program installations:

- Restrictions on system changes
- Privileges required to install
- Scope of their operation
- Size limitations
- Ability to query user for configuration information

On-demand security programs should not make any permanent changes to the unmanaged device. As soon as a protected session terminates, the program should remove itself completely from the system. To that end, executables should not be saved to disk and there should be no modifications to the registry.

On-demand security programs should not require any administrative privileges to install or operate. Many users, especially in corporate and government organizations, do not have administrator rights to install software on their machines. On-demand security should assume a minimal set of access rights to a device. This setup will limit the program to accessing only generally available information about the device, the OS, and unprotected data, such as cookies.

On-demand security programs must localize their function. They should control processes only related to the protected data. For example, if an on-demand security mechanism is downloaded to protect an online banking session, it should not interfere with another session that is downloading music files. Security operations are limited to the processes that are manipulating the protected data.

You can imagine situations in which it would be helpful to spread the protective measures to other processes. For example, if a spyware application is intercepting keyboard messages from the OS message queue searching for usernames and passwords, it would help the user if all browser sessions were protected from this monitoring. However, if a legitimate program—for example, a debugging or audit program—is intercepting the messages, the on-demand security program would essentially disable the other program.

On-demand security programs should function as depicted in Figure 5.10. Although all messages may pass through the same logical path in the OS, the security program should focus only on one application. To be precise, it should focus on one instance of an application. Although Figure 5.11 uses the labels Protected Application and Unprotected Application, in some cases, they could be two instances of the same program running at the same time.

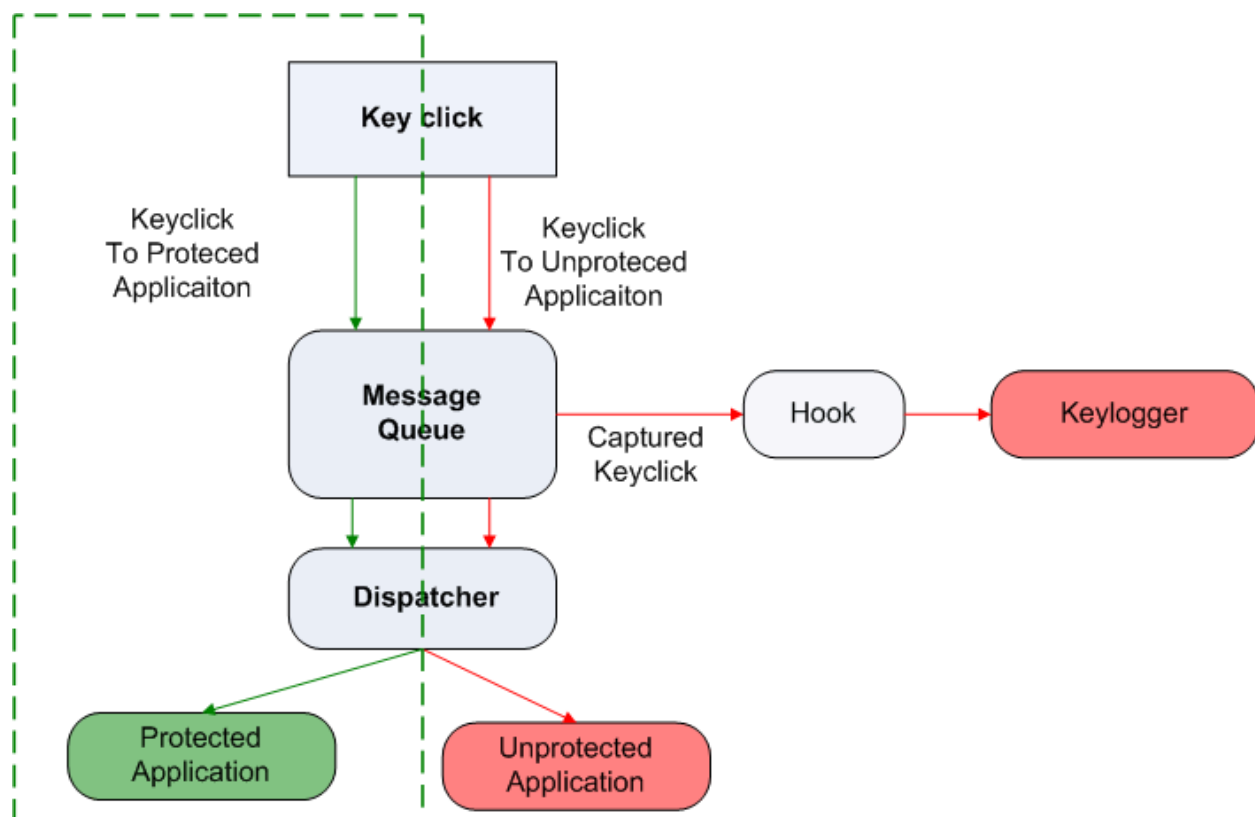


Figure 5.11: Applications protected with on-demand security localize protection to one application and should not interfere with other applications.

In addition to being selective about what countermeasures they take and on whose behalf they take them, these programs must be small. Downloading large, on-demand programs is not practical. They will take too long to download and will utilize large amounts of memory, possibly resulting in degraded performance as the unmanaged device is forced to page to virtual memory. The growing popularity and functionality of handheld mobile devices will likely increase the importance of keeping on-demand countermeasures lightweight.

Converging Devices: Mobile Devices and the Need for On-Demand Security

Desktops, laptops, personal digital assistants (PDAs), and smart phones are now serving overlapping functions. With the release of Windows Mobile by Microsoft and the availability of smart phones to run that OS, cell phones can now be used for tasks that we once did only at our desks:

- Track contacts
- Keep a schedule of appointments
- Track task lists
- Access email
- Browse the Web
- Work on word processing and spreadsheet documents

Functionality is also converging within mobile devices; a phone can now be a camera and a global positioning satellite client.

Network administrators and security managers might feel like they have enough to handle with wired networks, wireless access, VPNs, and handheld email devices. Smart phones only add to the list of emerging technologies that carry with them technology-specific risks. The OS, for example, may be in the Windows family, but it is different from the OS that runs on desktops. These devices will carry with them new sets of vulnerabilities and will become the target of new types of threats as well as variations on existing threats.

Smart phones are not immune to malware. The first cell phone worm, Cabir, was discovered in June 2004 on phones running the Symbian OS. In September 2005, the first cross-over malware, known as Cardtrap.A, was discovered. It attempts to infect PCs when users insert infected memory cards into their PCs.

The use of smart phones and similar devices constitutes a growing sub-group of unmanaged devices that will be accessing corporate networks, financial institution Web sites, self-service portals, healthcare services, and other areas with confidential information. Mobile devices will need on-demand security as much as, if not more so, than their stationary counterparts.

Protecting Information on Unmanaged Devices

When the several of the conditions just discussed are met, on-demand security measures can be deployed as needed. Once deployed, what functions can they carry out? After all, these are relatively small programs with no administrative rights to the device. In spite of the limitations, on-demand security measures can perform three key operations:

- Assess the security profile of the unmanaged device
- Disable application features based on the security profile
- Digitally shred data and remove on-demand security programs when the session terminates

These three key functions are the foundation for extending the trusted zone of the managed network to include unmanaged devices.

Assessing Security Profiles

The security profile of a device is the state of all programs and security controls on a device as well as vulnerabilities. These include:

- OS running on the device
- Patch level of the OS and major applications
- Configuration of network services
- State of background system services
- Status of countermeasures, such as antivirus and personal firewalls

Based on the findings of the assessment operation, on-demand security mechanisms can take appropriate measures. At the most basic level, on-demand security could check for the existence of antivirus software and personal firewalls, which, if not found, would cause the session to terminate. This type of “all-or-nothing” method is suitable for extreme cases but leaves no room for network administrators to enforce fine-grained policies.

A better approach is to examine the full state of the machine and take targeted action. For example, the on-demand security program could check that the latest service packs are installed. If not, the client device might be vulnerable to a worm that exploits a vulnerability in the ftp process, so using ftp during the session is disabled.

Checks can be more thorough as well. For example, rather than passing a device just because it has up-to-date antivirus and firewall software, on-demand security policies could dictate restrictions on services that have known vulnerabilities. Of course, administrative rights might be needed to retrieve detailed information about other system processes; however, vulnerabilities that can be exploited by malware can also be detected by vulnerability scanners embedded in on-demand security mechanisms.

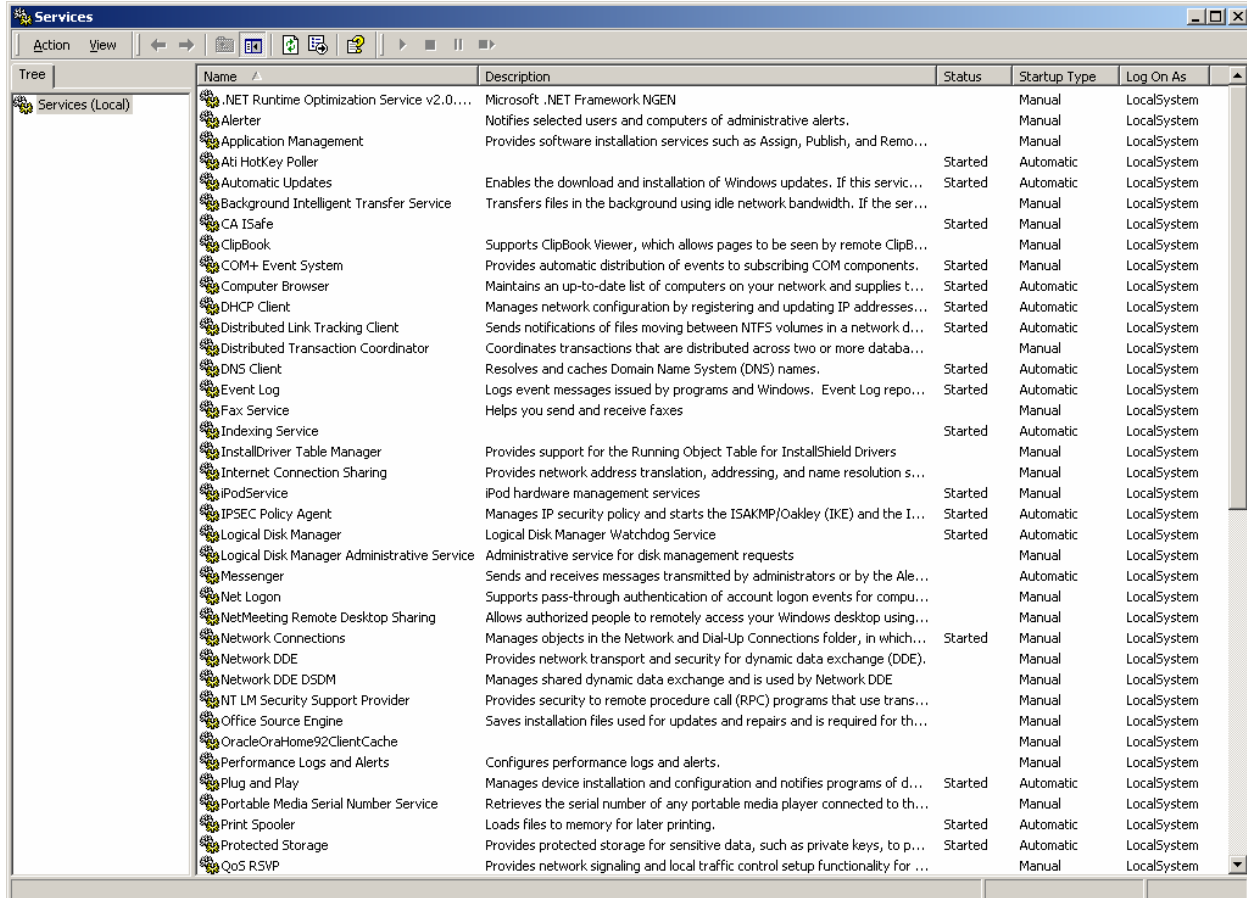


Figure 5.12: Users may be unaware of the many system services that run in the background of their computers.

Disabling Features based on Security Profile

As noted earlier, security on demand should not be limited to an “all-or-nothing” approach. Depending on the security profile of an unmanaged device, on-demand security should be able to control features available to the user. Three areas, in particular, present vulnerabilities to information on unmanaged devices:

- Printing
- Saving to disk
- Buffering data to a cache

By controlling the flow of information to these three functions, on-demand security can significantly increase the level of protection around data.

Printing Controls

Printing sounds relatively benign, but it can be a point of information loss. Consider a salesperson at a tradeshow who prints a report from SalesForce.com. Prior to starting the print job, the data is placed in the print queue and then sent to the printer. In the print queue, the information must be decrypted, so it is vulnerable to malware that can read the print queue.

Not all vulnerabilities are technical. That same salesperson may receive an urgent call while waiting for the printer and leave before the report is done. Something as simple as leaving a paper copy around can leak as much data as a sophisticated piece of malware on a vulnerable device.

Printing policies used in on-demand security should take into account:

- The user who is trying to print—Executives have access to detailed and valuable information that can warrant more stringent controls than is placed on others.
- The content that is printed—Email printing may be allowed but spreadsheet printing may be blocked under the assumption that spreadsheets are likely to contain confidential figures.
- The application in use—Users may be able to print from their calendars but are blocked when they direct their browser to a sales-tracking system or the corporate ERP.

Like printing, saving to disk is a common operation that should be controlled.

Limiting Save Options

Saving data to the disk on an unmanaged device is obviously risky. Many public use computers are configured to delete user data when the user logs off, but this setting depends on the features offered by the computer and assumes two things. First, that the feature actually works. Was this particular device configured properly? Does the site hosting this computer (for example, a hotel and conference center) have the IT staff necessary to ensure the configurations are maintained? What policies and procedures do they follow to ensure the integrity of their devices? Second, even if the data-deleting functions work correctly, the device could be compromised by malware that can access the saved data.

Buffer Caching

Data will be saved to buffer caches. Browsers depend on buffer caches to maintain acceptable performance. As the data in buffer caches is easily captured, see Figure 5.6 for an example, the data in the buffer should be encrypted and decrypted when used by the browser. At the very least, data in the browser caches should be deleted at the end of the session. Some browsers, such as Mozilla Firefox, provide users with this option (see Figure 5.13), but users must configure the browser to perform this operation.

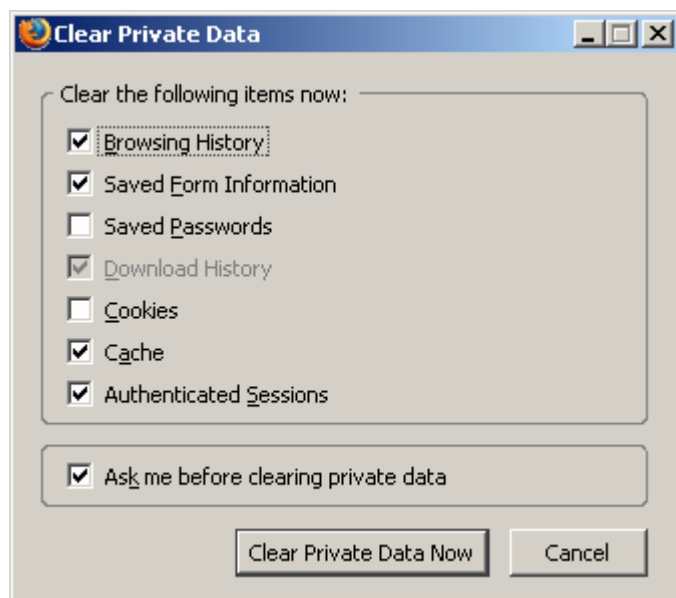


Figure 5.13: Browsers are offering better controls over deleting private data.

Terminating Sessions

When a session terminates, on-demand security programs should remove traces of all data that has been downloaded as well as any traces of the on-demand security program itself. Clearing caches, deleting any data saved to disk (if it were allowed by the security policy), and clearing the clipboard are the types of general data housekeeping that must be performed.

Summary

Data protection can no longer depend upon protecting the corporate servers and networks that house the data. The Web has redefined how users access information, and security measures must be redefined as well. The advent of on-demand security allows protective measures to be associated with data itself, not just the device that may house it at the moment. Confidential and valuable information moves with the people that conduct business and goes wherever they conduct business. Security must be as flexible as the data access and the on-demand security model is becoming another addition to the set of measures deployed by well-secured organizations.

Download Additional eBooks from Realtime Nexus!

Realtime Nexus—The Digital Library provides world-class expert resources that IT professionals depend on to learn about the newest technologies. If you found this eBook to be informative, we encourage you to download more of our industry-leading technology eBooks and video guides at Realtime Nexus. Please visit <http://nexus.realtimepublishers.com>.