CHAPTER 5

Live Data Collection from Windows Systems

One of the first steps of any preliminary investigation is to obtain enough information to determine an appropriate response. The steps you take to confirm whether or not an incident occurred vary depending on the type of incident. Obviously, you will take different steps to verify unacceptable web surfing than you will to determine whether an employee has been stealing files from another system's file shares. You need to take into consideration the totality of the circumstances before responding at the target system, using the standard investigative techniques outlined in Chapter 4. If we could become a broken record, we would repeat "totality of the circumstances" over and over. Initial response is an investigative as well as a technical process!

The goal of an initial response is twofold: Confirm there is an incident, and then retrieve the system's volatile data that will no longer be there after you power off the system. During your initial, hands-on response, you want to perform as few operations as possible to gather enough information to make the decision whether the incident warrants forensic duplication.

In this chapter, we outline the steps to take when performing the initial response to a Windows NT/2000/XP system, whether the system was used by an attacker or was the victim of an attack. We begin by discussing the pre-incident preparation and the creation of a response toolkit. Then we describe how to gather volatile data in a manner that minimizes the alteration of the system. Finally, we address making a decision about performing a forensic duplication of the evidence.

CREATING A RESPONSE TOOLKIT

For an initial response, you need to plan your approach to obtain all the information without affecting any potential evidence. Because you will be issuing commands with administrator rights on the victim system, you need to be particularly careful not to destroy or alter the evidence. The best way to meet this goal is to prepare a complete response toolkit.

CAUTION During severe incidents, you may have an audience of onlookers, gaping open-mouthed as you respond. Your response may be magic to them. These onlookers will be a distraction for you unless you are experienced, alert, and *prepared*.

Do not underestimate the importance of the monotonous and laborious step of creating a response toolkit. By spending the time to collect the trusted files and burn them onto a CD-ROM (or store them on floppies), you are much better equipped to respond quickly, professionally, and successfully. A live investigation is not the time to create or test your toolkit for the first time!

Gathering the Tools

In all incident responses, regardless of the type of incident, it is critical to use trusted commands. For responding to Windows, we maintain a CD or two floppy disks that contain a minimum of the tools listed in Table 5-1.

Tool	Description	Source
cmd.exe	The command prompt for Windows NT and Windows 2000	Built in
PsLoggedOn	A utility that shows all users connected locally and remotely	www.foundstone.com
`rasusers`	A command that shows which users have remote-access privileges on the target system	NT Resource Kit (NTRK)
`netstat`	A system tool that enumerates all listening ports and all current connections to those ports	Built in
Fport	A utility that enumerates all processes that opened any TCP/IP ports on a Windows NT/2000 system	www.foundstone.com
PsList	A utility that enumerates all running processes on the target system	www.foundstone.com
ListDLLs	A utility that lists all running processes, their command-line arguments, and the dynamically linked libraries (DLLs) on which each process depends	www.foundstone.com
`nbtstat`	A system tool that lists the recent NetBIOS connections for approximately the last 10 minutes	Built in
`arp`	A system tool that shows the MAC addresses of systems that the target system has been communicating with, within the last minute	Built in
`kill`	A command that terminates a process	NTRK
`md5sum`	A utility that creates MD5 hashes for a given file	www.cygwin.com
`rmtshare`	A command that displays the shares accessible on a remote machine	NTRK

Table 5-1. Response Toolkit Tools

Tool	Description	Source
`netcat`	A utility used to create a communication channel between two different systems	www.atstake.com/research/tools/network_utilities
`cryptcat`	A utility used to create an encrypted channel of communications	http://sourceforge.net/projects/cryptcat
PsLogList	A utility used to dump the contents of the event logs	www.foundstone.com
`ipconfig`	A system tool that displays interface configuration information	Built in
PsInfo	A utility that collects information about the local system build	www.foundstone.com
PsFile	A utility that shows files that are opened remotely	www.foundstone.com
PsService	A utility that shows information about current processes and threads	www.foundstone.com
`auditpol`	A utility used to display the current security audit settings	NTRK
`doskey`	A system tool that displays the command history for an open cmd.exe shell	Built in

Table 5-1. Response Toolkit Tools *(continued)*

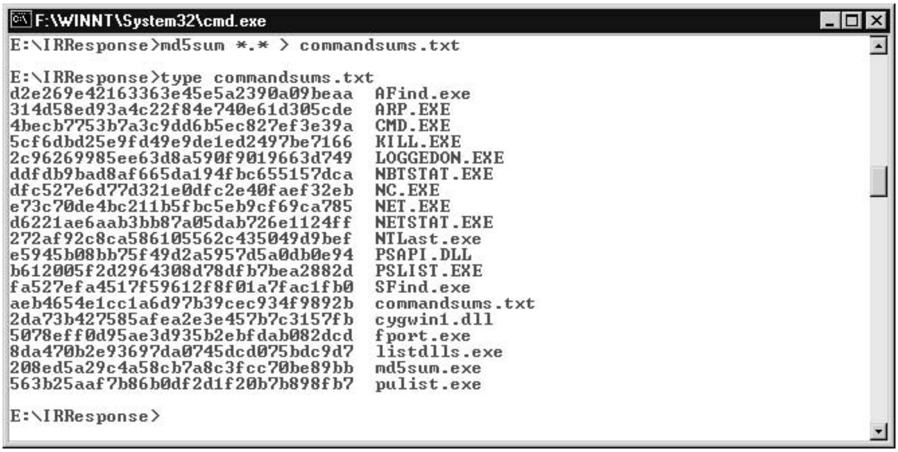
In Windows, there are two types of applications: those based on a graphical user interface (GUI) and those based on a console user interface (CUI). Since GUI programs create windows, have pull-down menus, and generally do "behind-the-scenes" interaction, we advise against using them for an investigation. All of the tools listed in Table 5-1 are CUI or command-line tools.

Preparing the Toolkit

You need to ensure that your toolkit will function exactly as intended and not alter the target system. We take several steps to prepare our toolkits for initial response:

- **Label the response toolkit media** A first step in evidence collection is to document the collection itself. Your response toolkit CD-ROM or floppy disks should be labeled to identify this part of your investigation. For example, for our response floppies and CDs, we make a specialized label that has the following information on it:

    - Case number
    - Time and date
    - Name of the investigator who created the response media
    - Name of the investigator using the response media
    - Whether or not the response media (usually a floppy disk) contains output files or evidence from the victim system
- **Check for dependencies with Filemon** It is important to determine which DLLs and files your response tools depend on. We use Filemon to determine all the files accessed and affected by each of the utilities in our toolkit. It is good to know which tools change access times on files on the target system. When we can, we avoid using "loud" tools that alter a lot of the target system.
- **Create a checksum for the response toolkit** One of the files on our response kit floppy (and CD and USB drive) is a text file with a checksum of all the commands on it. Figure 5-1 shows the `md5sum` command line used to create the text file (named commandsums.txt).
- **Write-protect any toolkit floppies** If you use floppy disks, be sure to write-protect the floppy after it is created. If you store evidentiary files on the response floppy during an incident, you need to write-protect it after you accumulate data and begin the chain of custody. The chain of custody tags should be filled out for each response floppy or CD, whether or not it contains evidence files. (See Chapter 9 for details on maintaining the chain of custody.)



```
E:\IRResponse>md5sum *.* > commandsums.txt

E:\IRResponse>type commandsums.txt
d2e269e42163363e45e5a2390a09beaa  AFind.exe
314d58ed93a4c22f84e740e61d305cde  ARP.EXE
4becb7753b7a3c9dd6b5ec827ef3e39a  CMD.EXE
5cf6dbd25e9fd49e9de1ed2497be7166  KILL.EXE
2c96269985ee63d8a590f9019663d749  LOGGEDON.EXE
ddfdb9bad8af665da194fbc655157dca  NBTSTAT.EXE
dfc527e6d77d321e0dfc2e40faef32eb  NC.EXE
e73c70de4bc211b5fbc5eb9cf69ca785  NET.EXE
d6221ae6aab3bb87a05dab726e1124ff  NETSTAT.EXE
272af92c8ca586105562c435049d9bef  NTLast.exe
e5945b08bb75f49d2a5957d5a0db0e94  PSAPI.DLL
b612005f2d2964308d78dfb7bea2882d  PSLIST.EXE
fa527efa4517f59612f8f01a7fac1fb0  SFind.exe
aeb4654e1cc1a6d97b39cec934f9892b  commandsums.txt
2da73b427585afea2e3e457b7c3157fb  cygwin1.dll
5078eff0d95ae3d935b2ebfdab082dcd  fport.exe
8da470b2e93697da0745dcd075bdc9d7  listdlls.exe
208ed5a29c4a58cb7a8c3fcc70be89bb  md5sum.exe
563b25aaf7b86b0df2d1f20b7b898fb7  pulist.exe

E:\IRResponse>
```

Figure 5-1. Using md5sum to create a checksum for the response toolkit

STORING INFORMATION OBTAINED DURING THE INITIAL RESPONSE

During your initial response, you will gather a lot of information from the live system. We use the term *live* to refer to a system that is relevant to an investigation, whether it is the attacking system or the victim, and is currently powered on. Think of it as the crime scene before photos are taken and bodies are removed. You are operating in an untrusted environment, where the unexpected should be anticipated.

You have four options when retrieving information from a live system:

- Save the data you retrieve on the hard drive of the target system.
- Record the data you retrieve by hand in a notebook.
- Save the data you retrieve onto the response floppy disk or other removable media.
- Save the data you retrieve on a remote "forensic system" using `netcat` or `cryptcat`.

Saving data to the hard drive is undesirable because it modifies the system. Recording data by hand is not practical due to the volume of information. Floppy drives are usually not a great choice because the data will not fit on the floppy. Other removable, writable media with a larger capacity than a floppy would be ideal, but the victim system may not have a drive for such media. However, we are happy to report a new solution: the removable USB drive. These small devices, about the size of your thumb, provide fantastic storage capabilities (up into the gigabyte range) and can be used to store your toolkit as well as the collected data. These devices have drivers built in, so they will work with any computer that sports a USB port and Windows software. USB ports are fairly ubiquitous now, so we recommend obtaining a few of these devices for your response toolkit.

Despite the proliferation of USB ports, you still need to be able to save data across a network. We often choose `netcat` to transfer the information from the target system to a remote forensic workstation.

Transferring Data with netcat

`netcat` is a freely available tool that creates a channel of communication between hosts. We use it during initial response to create a reliable, TCP connection between the target system and the forensic workstation used for analysis. All that you need to use `netcat` is an IP address on the target network and a laptop system with enough storage space to retain the information you gather.

Using `netcat` allows you to transfer all the relevant system information and files you require to confirm whether or not an incident occurred. This technique of information gathering promotes two sound practices:

- It lets you get on and off the target system quickly.
- It allows you to perform an offline review of the information attained.

Figure 5-2 illustrates the process of using `netcat` during initial response.

To use `netcat`, you initiate a `netcat listener` on the forensic workstation and redirect all incoming data to a file. Figure 5-3 illustrates the forensic workstation listening for incoming connections on port 2222. It will write the information received on that port to a file called pslist.

On the target system, `netcat` is used to funnel the output to your response commands to the forensic workstation. The command line in Figure 5-4 runs pslist, sending the output of the command to the forensic workstation, at IP address 192.168.0.20.

When transferring files in this manner, `netcat` does not know when the data transfer is complete. You will need to break the connection after the data transfer is complete by pressing CTRL-C on the forensic workstation. You will know data transfer is complete when the floppy or CD-ROM stops spinning on the target system or when the file size is no longer growing on the forensic workstation.

Use md5sum to Ensure Evidence Integrity

Remember to protect the integrity of the files you retrieve during the response using `md5sum`. We prefer to run `md5sum` on the files stored on the forensic workstation. We perform an `md5sum` in the presence of witnesses. We call it the two-man integrity rule.

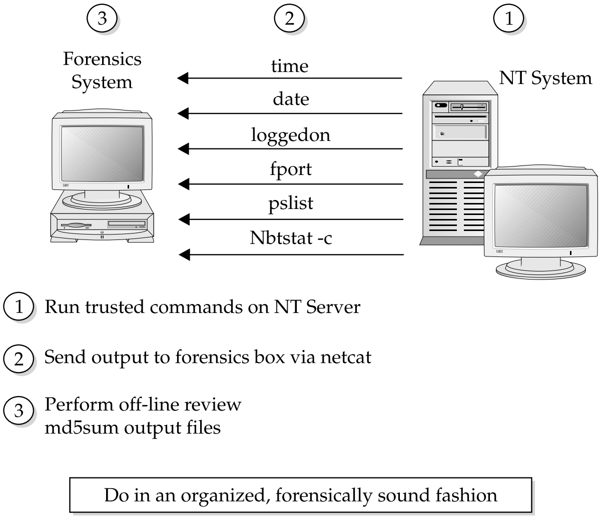
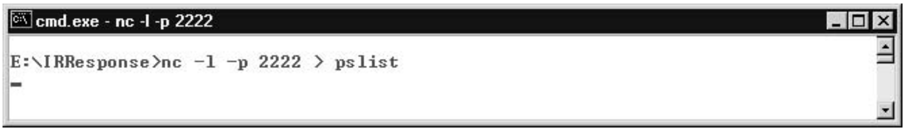

3
2
1
Forensics System
time
date
loggedon
fport
pslist
Nbtstat -c
NT System
1 Run trusted commands on NT Server
2 Send output to forensics box via netcat
3 Perform off-line review
md5sum output files
Do in an organized, forensically sound fashion

Figure 5-2. Using netcat during initial response to incidents



```
E:\IRResponse>nc -l -p 2222 > pslist
_
```

Figure 5-3. Setting up the netcat listener on the forensic workstation

GO GET IT ON THE WEB

netcat: http://www.atstake.com/research/tools/network_utilities
md5sum: http://www.cygnus.com

Encrypting Data with cryptcat

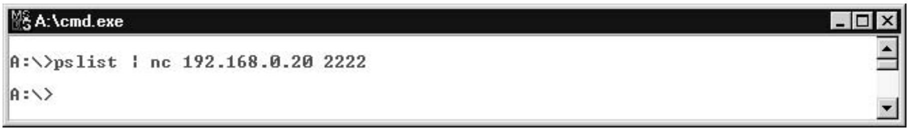
The drawback of transferring data across a network is that the data may be visible to network eavesdroppers. Consider encrypting the traffic using `cryptcat`. An alternative is to use a crossover cable to directly connect the victim system and the forensics workstation.

`cryptcat` has the same syntax and functions as the `netcat` command, but the data transferred is encrypted. There are two compelling arguments for encrypting your traffic when sending files from a target system:

- An attacker's sniffer cannot compromise the information you obtain.
- Encrypting the data nearly eliminates the risk of contamination or injection of data.

GO GET IT ON THE WEB

cryptcat: http://sourceforge.net/projects/cryptcat



```
A:\>pslist | nc 192.168.0.20 2222
A:\>
```

Figure 5-4. Sending the output of pslist to the forensic workstation

OBTAINING VOLATILE DATA

Now that you have a forensic toolkit and a methodology, you need to determine exactly which data to collect. At this point, you want to obtain the volatile data from the Windows NT/2000 system prior to turning off that system. At a minimum, we collect the following volatile data prior to forensic duplication:

- System date and time
- A list of the users who are currently logged on
- Time/date stamps for the entire file system
- A list of currently running processes
- A list of currently open sockets
- The applications listening on open sockets
- A list of the systems that have current or had recent connections to the system

If you know that your investigation is unlikely to require forensic duplication, you may want to collect more data. For example, you may want to dump RAM, obtain some information from the Registry, or perform other actions on the target system, pending the totality of the circumstances. Gathering this information is covered in the "Performing an In-Depth Live Response" section later in this chapter. Here, we describe the steps necessary to obtain critical data that is lost if you simply turn off the system and perform forensic duplication.

Consider the Best Time for an Incident Response

Carefully determine the most appropriate time to respond to the incident. If an employee is suspected of unacceptable use of his system to run an illicit business on company time and company resources, there may not be exigent circumstances that warrant immediate action, in broad daylight, in front of all the other employees. If you conduct your initial response at night or during a weekend, your actions will be more discreet. On the other hand, an active attack against your e-commerce server may warrant immediate action. The bottom line: Plan your response for the appropriate time.

Organizing and Documenting Your Investigation

It's one thing to have the technical skills required for proper incident response; it is quite another to implement a complete, unbiased, professional process. You need to have a methodology that is both organized and documented. There are two reasons for diligently documenting your actions when responding at the console of a victim system:

- To gather information that may become evidence against an individual
- To protect your own organization

What if the server you are retrieving information from crashes, and a client or your boss blames your actions for the downtime? If you dutifully documented your actions, you will have a written history of the steps you took on the machine, which should provide a defense to any challenge.

Before you begin collecting data, you should have an MD5 sum file with the checksums of each tool you will use. If you need to use untrusted binaries during a response, be sure to record the full pathnames of those binaries.

We recommend that you use a form to plan and document your response. For our investigations, we record the start time of the command executed and the command line entered. We document whether we ran a trusted or untrusted binary. Then we generate an MD5 sum of the data obtained by each command and add any relevant comments. Here is an example of such a form:

Start Time	Command Line	Trusted	Untrusted	MD5 Sum of Output	Comments
12:15:22	type lmhosts | nc 192.168.0.1 2222	X		3d2e531d.6553ee93e0890091. 3857eef3	Contents of lmhosts file
12:15:27	pslist | nc 192.168.0.1 2222	X		1ded672ba8b2ebf5beef672201 003fe8	
12:15:32	netstat –an | nc 192.168.0.1 2222	X		52285a2311332453efe20234385 7eef3	

Use a form like this to write down all the commands you are going to run before you respond on the target system. This approach ensures that the investigator plans ahead!

Legal Issues

It is a good idea to have a witness sign the form and verify each MD5 sum performed during the response. At the end of your response, before you review the output, copy all the output files and their corresponding checksums to backup media. Immediately provide copies to another party. Remember the two-man integrity rule!

Collecting Volatile Data

Now that you know what to collect and how to document your response, you are ready to retrieve the volatile data. We have created a "top-ten" list of the steps to use for data collection:

1. Execute a trusted cmd.exe.
2. Record the system time and date.
3. Determine who is logged in to the system (and remote-access users, if applicable).

4. Record modification, creation, and access times of all files.
5. Determine open ports.
6. List applications associated with open ports.
7. List all running processes.
8. List current and recent connections.
9. Record the system time and date.
10. Document the commands used during initial response.

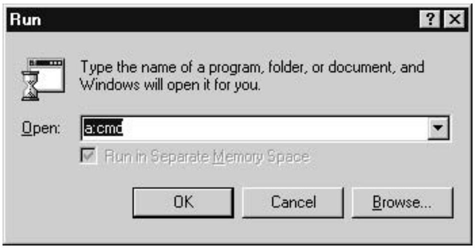
The following sections describe how to perform each of these steps. Remember that you may need to collect more data than what we show in this list.

Executing a Trusted Cmd.exe

As discussed in previous chapters, investigators need to be careful of tripwires or booby traps that attackers put in place to foil incident response. You may run what you think is cmd.exe on a victim system, only to discover that you actually executed `del *.*` in the \WINNT\System32 directory, rendering the system virtually inoperable. The solution is to execute a trusted version of cmd.exe from your own toolkit. Figure 5-5 illustrates using the Start | Run command on a Windows system to open a trusted cmd.exe on the floppy drive.

Recording the System Time and Date

After executing the trusted command shell, it is a good idea to capture the local system date and time settings. This is important to correlate the system logs, as well as to mark the times at which you performed your response. The `time` and `date` commands are a part of the cmd.exe application. Figure 5-6 illustrates the execution of the `date` command, redirecting the output to a file called date.txt on the floppy drive. The second command in the figure uses the append operator (>>) to add the output to the `time` command to the date.txt file.

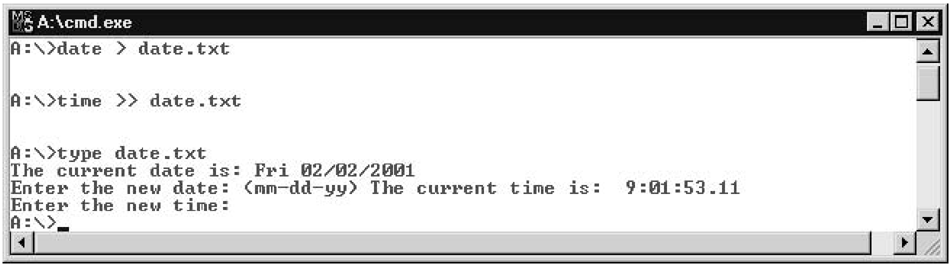


Type the name of a program, folder, or document, and Windows will open it for you.

Open: a:cmd

☑ Run in Separate Memory Space

Figure 5-5. Running a trusted version of cmd.exe



```
A:\>date > date.txt


A:\>time >> date.txt


A:\>type date.txt
The current date is: Fri 02/02/2001
Enter the new date: (mm-dd-yy) The current time is:  9:01:53.11
Enter the new time:
A:\>_
```

Figure 5-6. Obtaining the system time and date

When you execute the `date` and `time` commands, you must press the ENTER key to indicate that you do not want to change the settings.

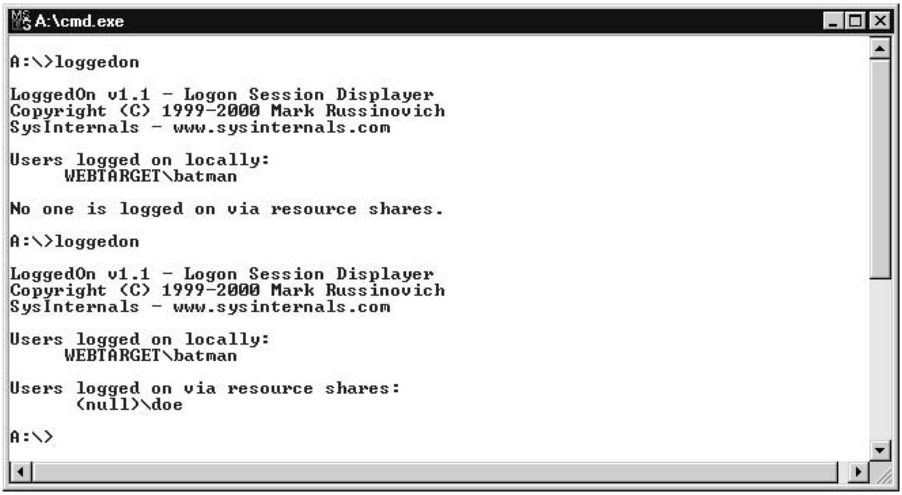
Be Consistent with Your Output Filenames

Maintain a consistent naming convention for your output files, such as prepending each command with a case and system identifier. An example would be the case number followed by the machine IP address, then a descriptive title, as in "FS030503-10.48.73.21-date.txt." Also, as soon as you create a file, immediately generate an MD5 sum of the results. This helps to ensure the integrity of the document file.

Determining Who Is Logged in to the System and Remote-Access Users

The next step is to determine which user accounts have active connections to the system. You want to know whose service you may be interrupting should you decide to terminate the network connections to the victim system. Mark Russinovich created PsLoggedOn, a utility that shows all users connected locally and remotely. Notice the null session connection from a remote system in Figure 5-7.

If you are responding to a system that offers remote access via modem lines, you need to determine which user accounts have remote-access privileges on the target system. If none do, you know that the modem is for outgoing connections (or at least not Remote Access Service, or RAS). If several accounts can access the system via RAS, you need to decide whether or not you want to pull the telephone lines from the system during the response. You may not want to allow any access to the target system while you are responding. The command-line tool to enumerate the users who can log in to a system via RAS is called `rasusers`.



```
A:\>loggedon

LoggedOn v1.1 - Logon Session Displayer
Copyright (C) 1999-2000 Mark Russinovich
SysInternals - www.sysinternals.com

Users logged on locally:
     WEBTARGET\batman

No one is logged on via resource shares.

A:\>loggedon

LoggedOn v1.1 - Logon Session Displayer
Copyright (C) 1999-2000 Mark Russinovich
SysInternals - www.sysinternals.com

Users logged on locally:
     WEBTARGET\batman

Users logged on via resource shares:
      (null)\doe

A:\>
```

Figure 5-7. Using PsLoggedOn to list users currently logged in to a system

Recording Modification, Creation, and Access Times of All Files

Use the `dir` command to get a directory listing of all the files on the target system, recording their size, access, modification, and creation times. *This is often the most important and critical step to incident response!*

If you can identify the relevant timeframe when an incident occurred, the time/date stamps become the evidence of which files an attacker touched, uploaded, downloaded, and executed. Windows performs this task extremely quickly. Here are examples of using `dir` to obtain access (`a`), modification (`w`), and creation (`c`) times:

`dir /t:a /a /s /o:d c:\`	Provides a recursive directory listing of all the access times on the C: drive
`dir /t:w /a /s /o:d d:\`	Provides a recursive directory listing of all the modification times on the D: drive
`dir /t:c /a /s /o:d e:\`	Provides a recursive directory listing of all the creation times on the E: drive

Determining Open Ports

To determine which ports are open, use `netstat`, a standard Windows command that enumerates all listening ports and all current connections to those ports. `netstat` is useful for recording volatile data such as current connections and connections that have just terminated. Figure 5-8 shows `netstat` being executed on an NT Server machine.

You will notice many localhost connections listed in the output. Even though a software package runs on a single machine, it may have been written with the client/server model in mind. Thus, `netstat` will almost always show connections between applications on the localhost 127.0.0.1. These connections are rarely of concern to the investigator. You will be looking for suspicious remote IP addresses and listening ports.

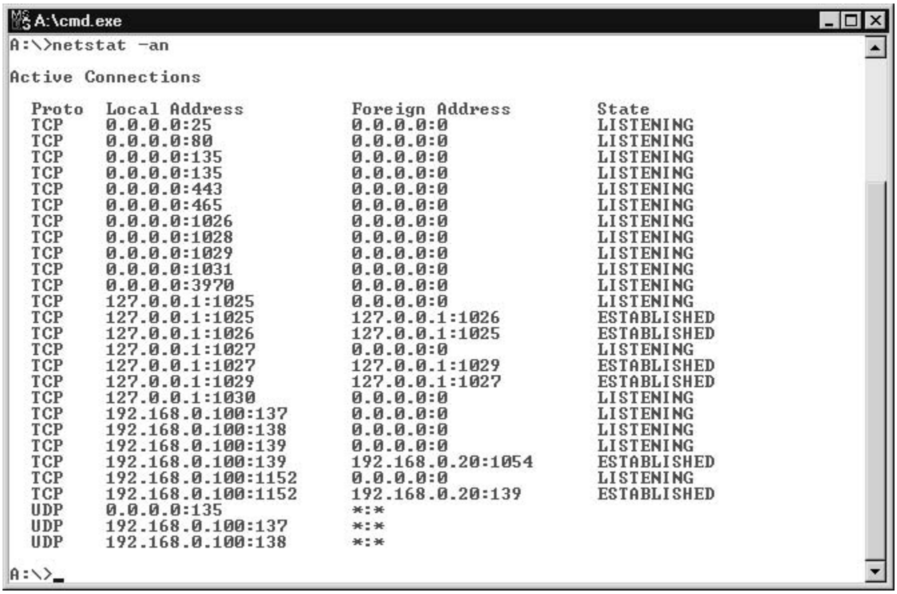
Listing Applications Associated with Open Ports

It is helpful to know which services listen on which specific ports. Otherwise, you will not be able to discern rogue processes from proper mission-critical processes. Foundstone supplies a free tool called Fport, which enumerates listening ports for all processes on a Windows NT/2000 system. Figure 5-9 shows the syntax for Fport and the corresponding output.

GO GET IT ON THE WEB

Fport: http://www.foundstone.com

If Fport reveals a rogue process listening for connections, and `netstat` shows current connections to that process, you may want to terminate the process to protect your



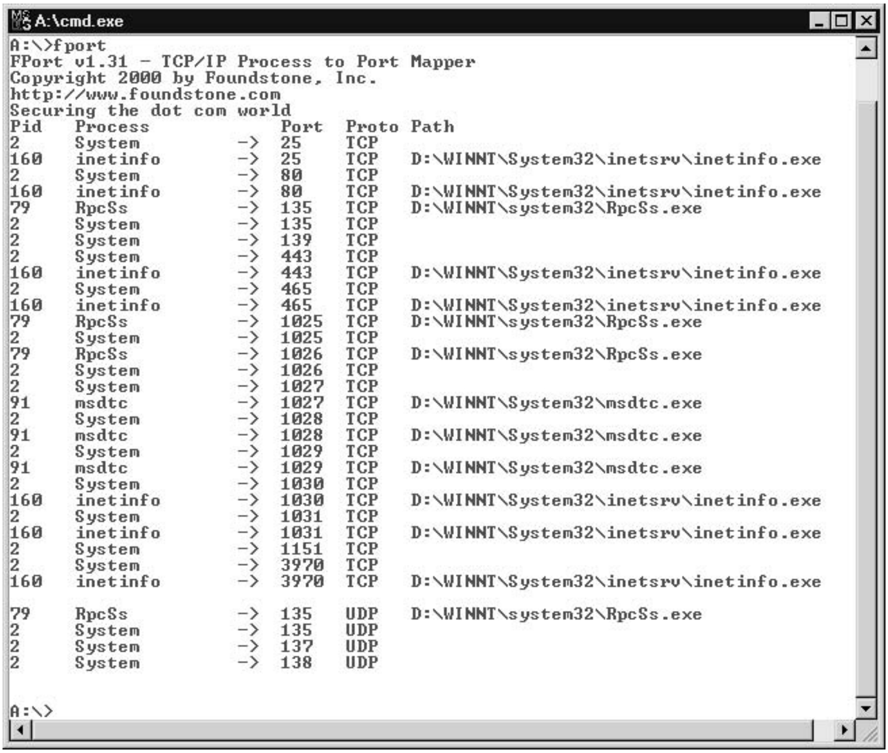
```
A:\>netstat -an

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:25             0.0.0.0:0              LISTENING
  TCP    0.0.0.0:80             0.0.0.0:0              LISTENING
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:443            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:465            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1026           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1028           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1029           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1031           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:3970           0.0.0.0:0              LISTENING
  TCP    127.0.0.1:1025         0.0.0.0:0              LISTENING
  TCP    127.0.0.1:1025         127.0.0.1:1026         ESTABLISHED
  TCP    127.0.0.1:1026         127.0.0.1:1025         ESTABLISHED
  TCP    127.0.0.1:1027         0.0.0.0:0              LISTENING
  TCP    127.0.0.1:1027         127.0.0.1:1029         ESTABLISHED
  TCP    127.0.0.1:1029         127.0.0.1:1027         ESTABLISHED
  TCP    127.0.0.1:1030         0.0.0.0:0              LISTENING
  TCP    192.168.0.100:137      0.0.0.0:0              LISTENING
  TCP    192.168.0.100:138      0.0.0.0:0              LISTENING
  TCP    192.168.0.100:139      0.0.0.0:0              LISTENING
  TCP    192.168.0.100:139      192.168.0.20:1054      ESTABLISHED
  TCP    192.168.0.100:1152     0.0.0.0:0              LISTENING
  TCP    192.168.0.100:1152     192.168.0.20:139       ESTABLISHED
  UDP    0.0.0.0:135            *:*
  UDP    192.168.0.100:137      *:*
  UDP    192.168.0.100:138      *:*

A:\>_
```

Figure 5-8. Using netstat to view current connections and listening ports



```
A:\>fport
FPort v1.31 - TCP/IP Process to Port Mapper
Copyright 2000 by Foundstone, Inc.
http://www.foundstone.com
Securing the dot com world
Pid   Process            Port  Proto Path
2     System         ->  25    TCP
160   inetinfo       ->  25    TCP   D:\WINNT\System32\inetsrv\inetinfo.exe
2     System         ->  80    TCP
160   inetinfo       ->  80    TCP   D:\WINNT\System32\inetsrv\inetinfo.exe
79    RpcSs          ->  135   TCP   D:\WINNT\system32\RpcSs.exe
2     System         ->  135   TCP
2     System         ->  139   TCP
2     System         ->  443   TCP
160   inetinfo       ->  443   TCP   D:\WINNT\System32\inetsrv\inetinfo.exe
2     System         ->  465   TCP
160   inetinfo       ->  465   TCP   D:\WINNT\System32\inetsrv\inetinfo.exe
79    RpcSs          ->  1025  TCP   D:\WINNT\system32\RpcSs.exe
2     System         ->  1025  TCP
79    RpcSs          ->  1026  TCP   D:\WINNT\system32\RpcSs.exe
2     System         ->  1026  TCP
2     System         ->  1027  TCP
91    msdtc          ->  1027  TCP   D:\WINNT\System32\msdtc.exe
2     System         ->  1028  TCP
91    msdtc          ->  1028  TCP   D:\WINNT\System32\msdtc.exe
2     System         ->  1029  TCP
91    msdtc          ->  1029  TCP   D:\WINNT\System32\msdtc.exe
2     System         ->  1030  TCP
160   inetinfo       ->  1030  TCP   D:\WINNT\System32\inetsrv\inetinfo.exe
2     System         ->  1031  TCP
160   inetinfo       ->  1031  TCP   D:\WINNT\System32\inetsrv\inetinfo.exe
2     System         ->  1151  TCP
2     System         ->  3970  TCP
160   inetinfo       ->  3970  TCP   D:\WINNT\System32\inetsrv\inetinfo.exe

79    RpcSs          ->  135   UDP   D:\WINNT\system32\RpcSs.exe
2     System         ->  135   UDP
2     System         ->  137   UDP
2     System         ->  138   UDP

A:\>
```

Figure 5-9. Using Fport to view listening services

system from potentially malicious actions taken by unauthorized intruders. When necessary, use the `kill` command to terminate rogue processes.

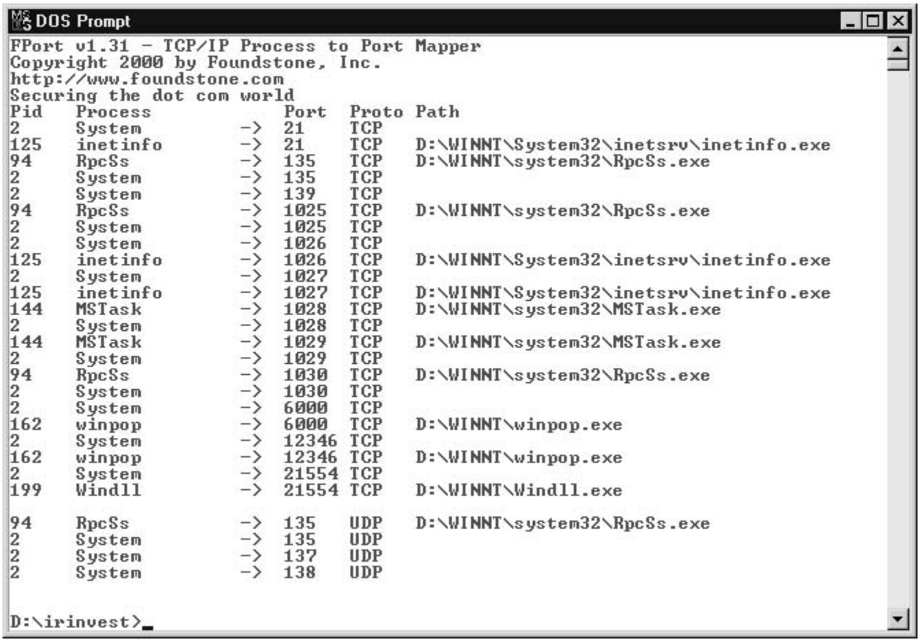
What Can Happen

You are sitting in front of your Windows NT system at work, when suddenly, your default web browser starts up and connects you to an online gambling site. You suspect that someone has installed some kind of remote-access server on your system.

Where to Look for Evidence

Figure 5-10 shows the results of running Fport on a system that has several remote-access trojans installed.

Process ID 162 looks suspicious, because \WINNT\winpop.exe is listening for connections on ports 6000 and 12346, which are ports commonly used by the popular Netbus trojan. Process ID 199 is also suspicious, because it shows Windll, which is used by the



```
FPort v1.31 - TCP/IP Process to Port Mapper
Copyright 2000 by Foundstone, Inc.
http://www.foundstone.com
Securing the dot com world
Pid   Process            Port  Proto Path
2     System        ->   21    TCP
125   inetinfo      ->   21    TCP   D:\WINNT\System32\inetsrv\inetinfo.exe
94    RpcSs         ->   135   TCP   D:\WINNT\system32\RpcSs.exe
2     System        ->   135   TCP
2     System        ->   139   TCP
94    RpcSs         ->   1025  TCP   D:\WINNT\system32\RpcSs.exe
2     System        ->   1025  TCP
2     System        ->   1026  TCP
125   inetinfo      ->   1026  TCP   D:\WINNT\System32\inetsrv\inetinfo.exe
2     System        ->   1027  TCP
125   inetinfo      ->   1027  TCP   D:\WINNT\System32\inetsrv\inetinfo.exe
144   MSTask        ->   1028  TCP   D:\WINNT\system32\MSTask.exe
2     System        ->   1028  TCP
144   MSTask        ->   1029  TCP   D:\WINNT\system32\MSTask.exe
2     System        ->   1029  TCP
94    RpcSs         ->   1030  TCP   D:\WINNT\system32\RpcSs.exe
2     System        ->   1030  TCP
2     System        ->   6000  TCP
162   winpop        ->   6000  TCP   D:\WINNT\winpop.exe
2     System        ->   12346 TCP
162   winpop        ->   12346 TCP   D:\WINNT\winpop.exe
2     System        ->   21554 TCP
199   Windll        ->   21554 TCP   D:\WINNT\Windll.exe

94    RpcSs         ->   135   UDP   D:\WINNT\system32\RpcSs.exe
2     System        ->   135   UDP
2     System        ->   137   UDP
2     System        ->   138   UDP

D:\irinvest>_
```

Figure 5-10. Recognizing unauthorized backdoors

GirlFriend trojan. One quick solution is to copy both files to the response floppy, and then use an up-to-date virus scanner or Pest Patrol on another system to determine if these programs are remote-access trojans. You could also get both winpop.exe (the Netbus trojan) and windll.exe (the GirlFriend trojan) for further analysis.

GO GET IT ON THE WEB

Trojan and remote access service ports: http://www.doshelp.com/trojanports.htm
Pest Data Catalog: http://research.pestpatrol.com/PestInfo/pestdatabase.asp

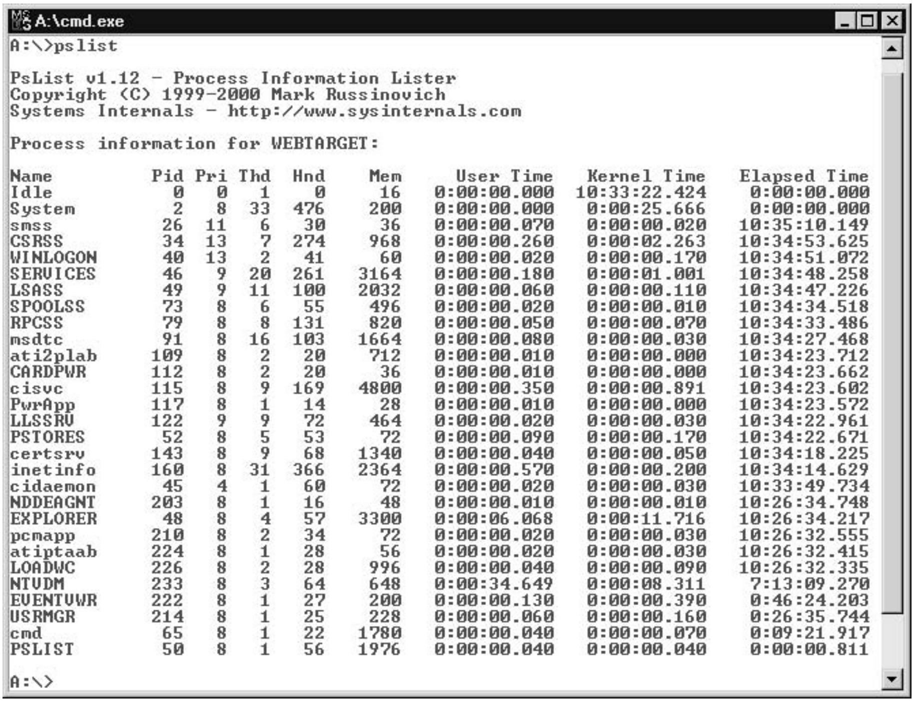
Listing All Running Processes

Before you power off a target system, it is important to record all of the processes currently running on that system. You cannot obtain this information if you simply unplug the power cord! When a process is executed on a Windows system, a kernel object and an address space that contains the executable code are created. The kernel object created is used by the operating system to manage the process and maintain statistical information about the process.

You can use Mark Russinovich's PsList utility to enumerate all running processes on the target system. Figure 5-11 shows an example of running PsList.

NOTE The original Windows API had no functions that enumerated the running processes from the kernel objects (no `ps` command as in Unix). The developers of Windows NT created the PSAPI.dll to enumerate which processes are running on a system. Windows 95 and 98 use a different API to enumerate processes, which we do not cover in this book.

If you cannot tell the difference between Windows critical processes and rogue processes, PsList will not be of much use to you. You need to recognize normal processes so that you can identify those processes that may be out of place or nefarious. For example, if PsList reveals that the EVENTVWR process is running, this suggests that someone is looking at the logs. If you see USRMGR, you might suspect that someone is trying to change the audit policies, add or delete a user account, or change user account data (passwords). Table 5-2 lists some common NT system processes.



```
A:\>pslist

PsList v1.12 - Process Information Lister
Copyright (C) 1999-2000 Mark Russinovich
Systems Internals - http://www.sysinternals.com

Process information for WEBTARGET:

Name           Pid Pri Thd  Hnd    Mem    User Time   Kernel Time   Elapsed Time
Idle             0   0   1    0     16  0:00:00.000  10:33:22.424    0:00:00.000
System           2   8  33  476    200  0:00:00.000   0:00:25.666    0:00:00.000
smss            26  11   6   30     36  0:00:00.070   0:00:00.020   10:35:10.149
CSRSS           34  13   7  274    968  0:00:00.260   0:00:02.263   10:34:53.625
WINLOGON        40  13   2   41     60  0:00:00.020   0:00:00.170   10:34:51.072
SERVICES        46   9  20  261   3164  0:00:00.180   0:00:01.001   10:34:48.258
LSASS           49   9  11  100   2032  0:00:00.060   0:00:00.110   10:34:47.226
SPOOLSS         73   8   6   55    496  0:00:00.020   0:00:00.010   10:34:34.518
RPCSS           79   8   8  131    820  0:00:00.050   0:00:00.070   10:34:33.486
msdtc           91   8  16  103   1664  0:00:00.080   0:00:00.030   10:34:27.468
ati2plab       109   8   2   20    712  0:00:00.010   0:00:00.000   10:34:23.712
CARDPWR        112   8   2   20     36  0:00:00.010   0:00:00.000   10:34:23.662
cisvc          115   8   9  169   4800  0:00:00.350   0:00:00.891   10:34:23.602
PwrApp         117   8   1   14     28  0:00:00.010   0:00:00.000   10:34:23.572
LLSSRV         122   9   9   72    464  0:00:00.020   0:00:00.030   10:34:22.961
PSTORES         52   8   5   53     72  0:00:00.090   0:00:00.170   10:34:22.671
certsrv        143   8   9   68   1340  0:00:00.040   0:00:00.050   10:34:18.225
inetinfo       160   8  31  366   2364  0:00:00.570   0:00:00.200   10:34:14.629
cidaemon        45   4   1   60     72  0:00:00.020   0:00:00.030   10:33:49.734
NDDEAGNT       203   8   1   16     48  0:00:00.010   0:00:00.010   10:26:34.748
EXPLORER        48   8   4   57   3300  0:00:06.068   0:00:11.716   10:26:34.217
pcmapp         210   8   2   34     72  0:00:00.020   0:00:00.030   10:26:32.555
atiptaab       224   8   1   28     56  0:00:00.020   0:00:00.030   10:26:32.415
LOADWC         226   8   2   28    996  0:00:00.040   0:00:00.090   10:26:32.335
NTVDM          233   8   3   64    648  0:00:34.649   0:00:08.311    7:13:09.270
EVENTVWR       222   8   1   27    200  0:00:00.130   0:00:00.390    0:46:24.203
USRMGR         214   8   1   25    228  0:00:00.060   0:00:00.160    0:26:35.744
cmd             65   8   1   22   1780  0:00:00.040   0:00:00.070    0:09:21.917
PSLIST          50   8   1   56   1976  0:00:00.040   0:00:00.040    0:00:00.811

A:\>
```

Figure 5-11. Using PsList to view all running processes

NT Process	Description
SMSS	The Session Manager that sets up the NT environment during the bootup process
CSRSS	The Client/Server Runtime Server Subsystem, used to maintain the Win32 system environment and numerous other vital functions
WINLOGON	The Windows logon service
SERVICES	Used by NT to manage services
LSASS	The Local Security Authority Security Service, which is always running to verify authentication on a system
SPOOLSS	The spooler service for the print subsystem
RPCSS	The remote procedure call subsystem
ATI2PLAB	A portion of the video driver subsystem
EXPLORER.EXE	Responsible for creating the Start button, desktop objects, and the taskbar
EVENTVWR	The Event Viewer application
USRMGR	The User Manager application
MSDTC	The Microsoft Distributed Transaction Coordinator, which is configured to start automatically when an NT system starts

Table 5-2. Some Windows NT System Processes

NOTE If you lose the Windows desktop for some reason (such as a hung process), you can choose Start | Run and enter **Explorer.** The desktop should reappear.

Listing Current and Recent Connections

`netstat`, `arp`, and `nbtstat` are useful utilities for determining who is connected or has recently connected to a system. Many Windows NT/2000 workstations have audit policies that do not log successful or failed logons. Therefore, these three utilities may be your only way to identify a remote system connecting to a workstation.

▼ `netstat` Many computer security specialists use `netstat` to list the open ports on a system. Since Fport lists the open ports and the application listening on each port, we use `netstat` to determine current connections and the remote IP addresses of those current connections, and to view recent connections.

- arp This utility is used to access the ARP cache, which maps the IP address to the physical MAC address for the systems that the target system has been communicating with in the last minute.
- nbtstat This utility is used to access the remote NetBIOS name cache, listing the recent NetBIOS connections for approximately the last ten minutes. Figure 5-12 shows an example of using nbtstat to list current and recent NetBIOS connections.

Recording the System Time and Date

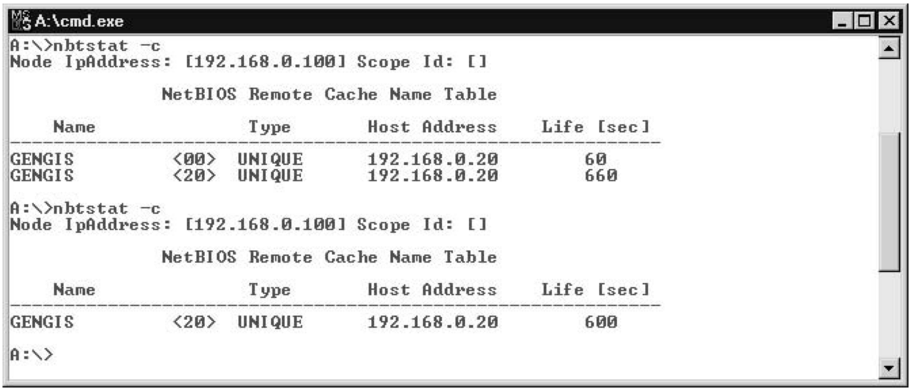
Issue the date and time commands again (repeat step 2) to record the time and date that you completed the live data collection. This ensures that you have a record of when you were on the system, so that if anything is changed on the system outside this timeframe, you will know that you are not responsible for the alteration.

Legal Issues

Sandwiching your data-retrieval commands between time and date commands is a forensically sound principle. This may become critical if an adversary challenges the steps you took during a response. You can pinpoint the exact actions you took on the system and the exact timeframe in which you took them.

Documenting the Commands Used during Initial Response

Use the doskey /history command to display the command history of the current command shell on a system (if the situation warrants). We also use doskey /history to



```
A:\>nbtstat -c
Node IpAddress: [192.168.0.100] Scope Id: []

              NetBIOS Remote Cache Name Table

    Name              Type       Host Address    Life [sec]
------------------------------------------------------------
GENGIS         <00>  UNIQUE      192.168.0.20        60
GENGIS         <20>  UNIQUE      192.168.0.20        660

A:\>nbtstat -c
Node IpAddress: [192.168.0.100] Scope Id: []

              NetBIOS Remote Cache Name Table

    Name              Type       Host Address    Life [sec]
------------------------------------------------------------
GENGIS         <20>  UNIQUE      192.168.0.20        600

A:\>
```

Figure 5-12. Using nbtstat to view recent NetBIOS connections

keep track of the commands executed on the system during a response, as shown in Figure 5-13.

Scripting Your Initial Response

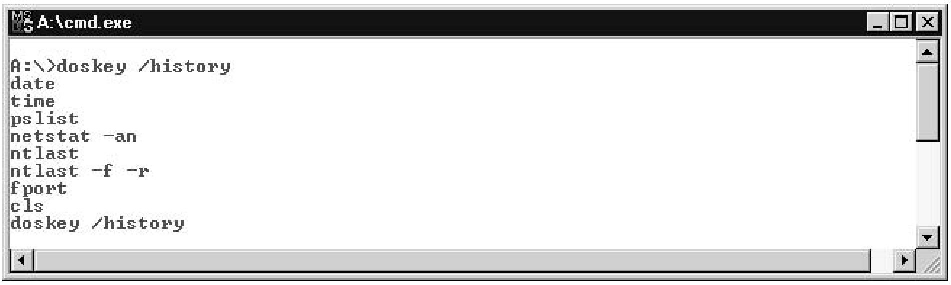
Many of the steps taken during the initial response can be incorporated into a single batch script. We often script our response, and then use `netcat` to transfer the results of the script to a forensic workstation. Simply create a text file and add a .bat extension to it to make it a batch file. Here is a sample script that can be used when responding to incidents on Windows NT/2000 systems:

```
time /t
date /t
psloggedon
dir /t:a /o:d /a /s c:\
dir /t:w /o:d /a /s c:\
dir /t:c /o:d /a /s c:\
netstat -an
fport
pslist
nbtstat -c
time /t
date /t
doskey /history
```

We named the above file ir.bat, and we run it on target systems to get the bare essentials. Notice how we surround the response with the `time` and `date` commands.

When redirecting the output of a script of multiple commands to a single `netcat` socket, you need to use the following command line on your analysis system:

```
nc.exe -L -p 2222 >> iroutput.txt
```



```
A:\>doskey /history
date
time
pslist
netstat -an
ntlast
ntlast -f -r
fport
cls
doskey /history
```

Figure 5-13. Using doskey to record the steps taken during incident response

The L stands for listen harder, telling the netcat socket not to close without user intervention (CTRL-C). The results are a single text file (iroutput.txt, in this case), which contains all the volatile information, recorded in a neat fashion.

PERFORMING AN IN-DEPTH LIVE RESPONSE

Sometimes, your response at the console of a live system needs to go beyond merely obtaining the volatile information. Perhaps shutting off the target system is not even an option, because there are numerous concerns about disruption of service.

You may need to find evidence and properly remove rogue programs without disrupting any services provided by the victim machine. In other words, you will not be able to shut off the machine, disable network connections, overtax the CPU, or use Safeback and EnCase (or any other popular Windows/DOS-based forensic software). This is somewhat contrary to traditional computer forensics, but the requirement to be able to retrieve forensically sound data without disrupting the operation of the victim computer is becoming more common.

CAUTION Unless you are experienced and know exactly how to pluck out all of the evidence needed during a live response, you should strongly consider forensic duplication of the victim system. In-depth live response should be left to the professionals who know exactly what to look for. Otherwise, you may be left with an incomplete response without evidence or proper purging of rogue processes and files.

Collecting the Most Volatile Data

Your first steps are to collect the most volatile data, as described in the previous sections and summarized here:

- Run the date and time commands to sandwich your response between starting and ending times. These commands record the current system time for correlation between system logs and network-based logging.
- Use PsLoggedOn to see who is currently connected to the system.
- Use netstat to view current and recent connections on all listening ports.
- Run PsList to see all the running processes.
- Use Fport to determine which programs have opened specific ports. If Fport indicates that a rogue process is running, obtain the rogue process for tool analysis.

After gathering this information, you can continue with some investigative steps that minimize the disruption of a target system's operation.

Creating an In-Depth Response Toolkit

To perform an in-depth live response, we use the tools listed in Table 5-3. These tools are presented in the order in which they are commonly used, but you may need to use them

Tool	Description	Source
`auditpol`	A command-line tool that determines the audit policy on a system	NTRK
`reg`	A command-line tool used to dump specific information (keys) within the NT/2000 Registry	NTRK
`regdump`	A command-line tool that dumps the Registry as a text file	NTRK
pwdump3e	A utility that dumps the SAM database so that the passwords can be cracked	www.polivec.com/pwdump3.html
NTLast	A utility that monitors successful and failed logons to a system	www.foundstone.com
Sfind	A utility that detects files hidden within NTFS file streams	www.foundstone.com
Afind	A utility that can search a file system to determine files accessed during specific timeframes	www.foundstone.com
`dumpel`	A command-line tool that is used to dump the NT/2000 event logs	NTRK

Table 5-3. Tools Used for an In-Depth Response

in a different order to meet the needs of your specific situation. Each one of these commands has standard output, which means that you can use all of these commands in conjuction with `netcat` to respond across a network connection.

Collecting Live Response Data

Two key sources of evidence on Windows NT/2000 systems are the event logs (if auditing is on) and the Registry on the target system. Thus, a thorough review of both is required during most investigations. We use the following approach, which obtains quite a bit of information from a live Windows NT/2000 system:

- Review the event logs.
- Review the Registry.
- Obtain system passwords.
- Dump system RAM.

These steps are discussed in more detail in the following sections.

Obtaining Event Logs during Live Response

Several tools are helpful for obtaining event logs from a live system:

- `auditpol` This tool from the NTRK discovers which audit policies exist on the system. Why try to obtain logs from a system if none exist? If Security Policy Changes auditing is turned on, you will find events recorded in the Security log (event ID 612). Figure 5-14 shows the command line and output for `auditpol`.
- NTLast Developed by Foundstone's J.D. Glaser, this is an excellent tool that allows you to monitor successful and failed logons to a system, if the system's Logon and Logoff auditing is turned on. You will want to look for suspicious user accounts and remote systems accessing the target system.
  - `ntlast` shows successful logons (Figure 5-15).
  - `ntlast -f` enumerates failed console logons (Figure 5-16).
  - `ntlast -r` lists all successful logons from remote systems (Figure 5-17).
  - `ntlast -f -r` shows failed remote logons.
- `dumpel` You will want to retrieve the other logs for offline analysis. Why search randomly on the target system using Event Viewer? Use `dumpel` (from the NTRK) and `netcat` to retrieve remote logs.
  - `dumpel -l security -t` dumps the entire Security log, with tabs as delimiters, to any file you specify.
  - `dumpel -l application -t` dumps the Application log to standard output.

GO GET IT ON THE WEB

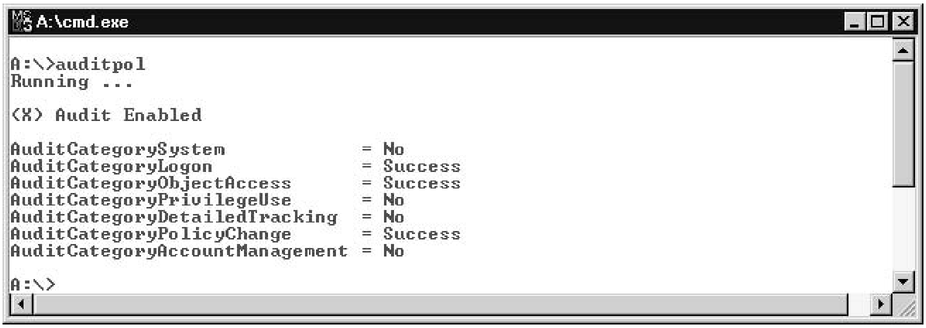
NTLast: http://www.foundstone.com

What Can Happen

An attacker sends an email with a remote-access trojan attachment to several recipients at an organization. The attacker is hoping that the recipients will unwittingly execute her trojan, allowing her backdoor access to the organization's network. However, the attacker's trojan fails to execute properly because the organization requires that every desktop system run an anti-virus program that quarantines evil files.

Where to Look for Evidence

The following entry is a victim system's Application log. Notice how the system HOMER4 was infected by a file called 04.D, which is actually the BackGate trojan. Also notice that this file was located in the c:\Inetpub\scripts directory. This file was probably placed on the system via a web server hack, such as the popular MDAC attack or the Internet Information Server (IIS) Unicode/double decode attack. The trojan was placed in the directory where



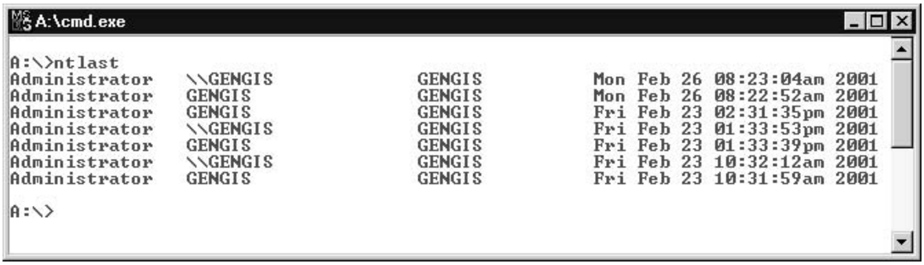
```
A:\>auditpol
Running ...

(X) Audit Enabled

AuditCategorySystem            = No
AuditCategoryLogon             = Success
AuditCategoryObjectAccess      = Success
AuditCategoryPrivilegeUse      = No
AuditCategoryDetailedTracking  = No
AuditCategoryPolicyChange      = Success
AuditCategoryAccountManagement = No

A:\>
```

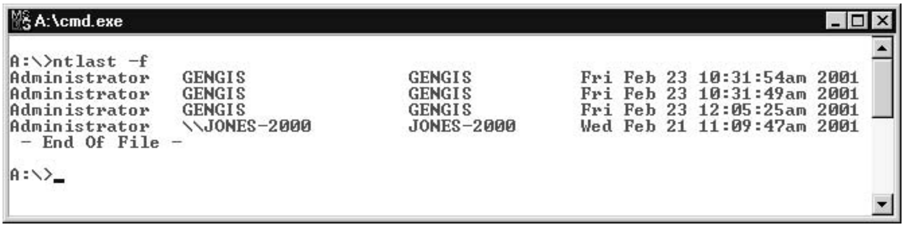
Figure 5-14. Using auditpol to determine system logging



```
A:\>ntlast
Administrator   \\GENGIS           GENGIS         Mon Feb 26 08:23:04am 2001
Administrator   GENGIS             GENGIS         Mon Feb 26 08:22:52am 2001
Administrator   GENGIS             GENGIS         Fri Feb 23 02:31:35pm 2001
Administrator   \\GENGIS           GENGIS         Fri Feb 23 01:33:53pm 2001
Administrator   GENGIS             GENGIS         Fri Feb 23 01:33:39pm 2001
Administrator   \\GENGIS           GENGIS         Fri Feb 23 10:32:12am 2001
Administrator   GENGIS             GENGIS         Fri Feb 23 10:31:59am 2001

A:\>
```

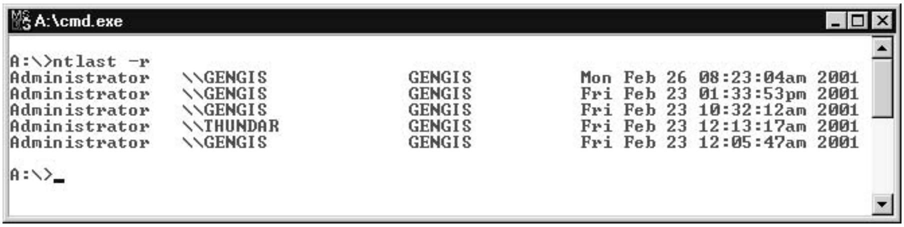
Figure 5-15. Using NTLast to view successful logons



```
A:\>ntlast -f
Administrator   GENGIS             GENGIS         Fri Feb 23 10:31:54am 2001
Administrator   GENGIS             GENGIS         Fri Feb 23 10:31:49am 2001
Administrator   GENGIS             GENGIS         Fri Feb 23 12:05:25am 2001
Administrator   \\JONES-2000       JONES-2000     Wed Feb 21 11:09:47am 2001
 - End Of File -

A:\>_
```

Figure 5-16. Using NTLast to view failed logons



```
A:\>ntlast -r
Administrator   \\GENGIS          GENGIS        Mon Feb 26 08:23:04am 2001
Administrator   \\GENGIS          GENGIS        Fri Feb 23 01:33:53pm 2001
Administrator   \\GENGIS          GENGIS        Fri Feb 23 10:32:12am 2001
Administrator   \\THUNDAR         GENGIS        Fri Feb 23 12:13:17am 2001
Administrator   \\GENGIS          GENGIS        Fri Feb 23 12:05:47am 2001

A:\>_
```

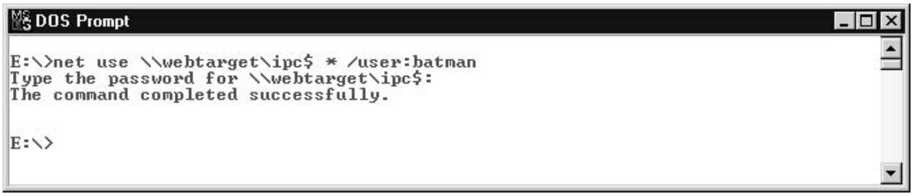
Figure 5-17. Using NTLast to view successful logons from remote systems

the default web server scripts are stored. It is likely that the attacker had placed an Active Server Pages (ASP) page that allowed her to upload arbitrary files.

```
3/4/03     3:38:43 PM  1     0     257     AlertManager
N/A     HOMER4     NetShield NT: The file C:\Inetpub\scripts\04.D
on HOMER4 is infected with the virus BackGate.  Unable to clean file.
  Cleaner unavailable or unable to access the file.
```

You can also view the logs on the target system remotely by choosing Log | Select Computer. You will need to have administrator-level access in order to remotely view the Security log on a remote system. Figure 5-18 illustrates how to establish a NetBIOS connection to the remote system to the IPC share, logging in to system `webtarget` as `batman` (which just happens to be the administrator account).

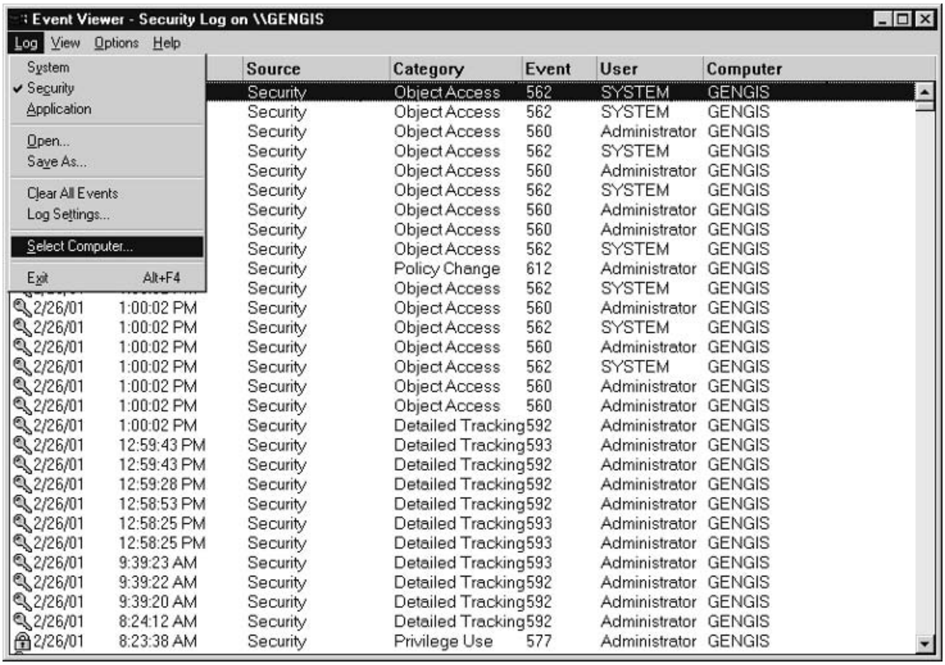
After you have the administrator account connection, simply choose Log | Select Computer, and you will be able to remotely view the event log on that system, as shown in Figure 5-19. If you want to create a local copy, save the file.



```
E:\>net use \\webtarget\ipc$ * /user:batman
Type the password for \\webtarget\ipc$:
The command completed successfully.

E:\>
```

Figure 5-18. Connecting to a remote system administrator account



		Source	Category	Event	User	Computer
		Security	Object Access	562	SYSTEM	GENGIS
		Security	Object Access	562	SYSTEM	GENGIS
		Security	Object Access	560	Administrator	GENGIS
		Security	Object Access	562	SYSTEM	GENGIS
		Security	Object Access	560	Administrator	GENGIS
		Security	Object Access	562	SYSTEM	GENGIS
		Security	Object Access	560	Administrator	GENGIS
		Security	Object Access	560	Administrator	GENGIS
		Security	Object Access	562	SYSTEM	GENGIS
		Security	Policy Change	612	Administrator	GENGIS
		Security	Object Access	562	SYSTEM	GENGIS
2/26/01	1:00:02 PM	Security	Object Access	560	Administrator	GENGIS
2/26/01	1:00:02 PM	Security	Object Access	562	SYSTEM	GENGIS
2/26/01	1:00:02 PM	Security	Object Access	560	Administrator	GENGIS
2/26/01	1:00:02 PM	Security	Object Access	562	SYSTEM	GENGIS
2/26/01	1:00:02 PM	Security	Object Access	560	Administrator	GENGIS
2/26/01	1:00:02 PM	Security	Object Access	560	Administrator	GENGIS
2/26/01	1:00:02 PM	Security	Detailed Tracking	592	Administrator	GENGIS
2/26/01	12:59:43 PM	Security	Detailed Tracking	593	Administrator	GENGIS
2/26/01	12:59:43 PM	Security	Detailed Tracking	592	Administrator	GENGIS
2/26/01	12:59:28 PM	Security	Detailed Tracking	592	Administrator	GENGIS
2/26/01	12:58:53 PM	Security	Detailed Tracking	592	Administrator	GENGIS
2/26/01	12:58:25 PM	Security	Detailed Tracking	593	Administrator	GENGIS
2/26/01	12:58:25 PM	Security	Detailed Tracking	593	Administrator	GENGIS
2/26/01	9:39:23 AM	Security	Detailed Tracking	593	Administrator	GENGIS
2/26/01	9:39:22 AM	Security	Detailed Tracking	592	Administrator	GENGIS
2/26/01	9:39:20 AM	Security	Detailed Tracking	592	Administrator	GENGIS
2/26/01	8:24:12 AM	Security	Detailed Tracking	592	Administrator	GENGIS
2/26/01	8:23:38 AM	Security	Privilege Use	577	Administrator	GENGIS

Figure 5-19. Using Event Viewer to review a remote system's event logs

NOTE We included how to access the event logs via a network connection solely because we are frequently asked how remote administration of Windows systems can be conducted. We do not feel that this is a sound methodology when responding to a computer security incident.

Reviewing the Registry during a Live Response

The Windows Registry stores a wealth of important data that is useful during initial response. We cover the full details of investigating the Registry in Chapter 12.

For live retrieval of the important Registry data, you can use `regdump` or `reg query`, both from the NTRK. The `regdump` utility creates an enormous text file of the Registry. We use `reg query` instead and extract just the Registry key values of interest. Here is a sample batch file that we have used to get some information from a target NT system:

```
REM To Get User Information
reg query "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\
RegisteredOwner"
```

```
reg query "HKLM\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\RegisteredOrganization"
reg query "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProductID"
reg query "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList"
reg query "HKLM\SAM\SAM\Domains\Account\Users\Names"
reg query "HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon"

REM To Get System Information
reg query "HkLM\SYSTEM\ControlSet001\Control\ComputerName\Computername"
reg query "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\CSDVersion"
REM To Get Banner Text If It Exists
reg query "HKLM\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Winlogon\LegalNoticeText"
REM To See If the Swap File Is Overwritten If the System Is Rebooted
 1=Yes 0=No
reg query "HKLM\System\CurrentControlSet\Control\Session Manager\Memory
Management\ClearPageFileAtShutdown"
Rem To See If the Admin Shares Are Shared on an NT Workstation 1=Shared
reg query "HKLM\System\CurrentControlSet\Services\LanmanServer\Parameters\
AutoShareWks"
REM To See Shares Offered on the System
reg query "HKLM\System\CurrentControlSet\Services\LanmanServer\Shares"

REM To Get Recent Files Used - Usually Needs Reconfiguring
reg query "HKCU\Software\MIcrosoft\Office\9.0\PowerPoint\RecentFileList"
reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\
RecentDocs"

REM To See All the Startup Programs
reg query "HKLM\Software\Microsoft\Windows\CurrentVersion\Run"
reg query "HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce"
reg query "HKLM\Software\Microsoft\Windows\CurrentVersion\RunServices"
reg query "HKLM\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce"
reg query "HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows\Load"
reg query "HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows\Run"
reg query "HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\
Userinit"
reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\Run"
reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce"
reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\RunServices"
reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce"
REM To See the Last Few Systems the Telnet Client Connected to
reg query "HKCU\Software\Microsoft\Telnet\LastMachine"
reg query "HKCU\Software\Microsoft\Telnet\Machine1"
reg query "HKCU\Software\Microsoft\Telnet\Machine2"
reg query "HKCU\Software\Microsoft\Telnet\Machine3"
```

You can tailor this example to get information about the Registry keys that are of interest on your system.

What Can Happen

An attacker uploads a remote-access trojan to the victim system and places an entry in the RunOnce key of the Registry so that the rogue application will be executed every time the victim system is rebooted.

Where to Look for Evidence

The following is a section of the Registry we retrieved from a victim system. You will note that two programs, windll.exe and winpop.exe, are executed each time the system is booted. The next step would be to obtain \WINNT\windll.exe and \WINNT\winpop.exe and perform tool analysis to determine their functions.

```
A:\>reg query "HKLM\Software\Microsoft\Windows\CurrentVersion\Run"
Listing of [Software\Microsoft\Windows\CurrentVersion\Run]
REG_SZ        SystemTray     SysTray.Exe
REG_SZ        BrowserWebCheck     loadwc.exe
REG_SZ        SchedulingAgent     mstinit.exe /logon
REG_SZ        AtiPTA     Atiptaab.exe
REG_SZ        WinPoET     c:\BANetDSL\WinPoET\WinPPPoverEthernet.exe
REG_SZ        Windll.exe     D:\WINNT\Windll.exe
REG_SZ        winpop     D:\WINNT\winpop.exe /nomsg:
[OptionalComponents]
```

Obtaining System Passwords

You may need to get the passwords off the system at the time of response, particularly if you have an uncooperative user. Use pwdump3e, an updated version of Todd Sabin's pwdump, to dump the passwords from the Security Accounts Manager (SAM) database. These passwords may be cracked on a forensic workstation using John the Ripper, L0phtcrack, or any other Windows password-cracking tool. Remember that if you decide to do a forensic duplication of the system, you will likely need the system passwords to boot the system into its native NT/2000 operating system. You will want to be able to log on with the administrator account.

GO GET IT ON THE WEB

pwdump3e: http://www.polivec.com/pwdump3.html

L0phtcrack: http://www.atstake.com/research/lc/index.html

John the Ripper: http://www.openwall.com/john

Dumping System RAM

It may be important for you to dump the contents of memory, perhaps to obtain passwords, get the cleartext of a recently typed encrypted message, or retrieve the contents of a recently opened file. Unfortunately, it is sometimes difficult to obtain Windows memory contents in a forensically sound manner.

There are two types of memory that an investigator may wish to obtain: user mode (application) memory and full-system memory. You can use the userdump.exe utility, part of the Microsoft OEM Support Tools package, to dump application memory. You can use George Garner's modified version of the GNU utility `dd` to dump full-system memory. For details on using these utilities, see Curtis Rose's whitepaper, "Windows Live Incident Response Volatile Data Collection: Non-disruptive User and System Memory Forensic Acquisition."

GO GET IT ON THE WEB

Microsoft support tools: http://download.microsoft.com/download/win2000srv/Utility/3.0/NT45/EN-US/Oem3sr2.zip

dd.exe for system memory collection: http://users.erols.com/gmgarner/forensics

IS FORENSIC DUPLICATION NECESSARY?

After reviewing the system information you retrieved during the initial response, you need to decide whether to perform a *forensic duplication* of the evidence. The forensic duplication of the target media provides the mirror image of the target system. This methodology provides due diligence when handling critical incidents. It also provides a means for having working copies of the target media for analysis without worrying about altering or destroying potential evidence. Generally, if the incident is severe or deleted material may need to be recovered, a forensic duplication is warranted.

Law enforcement generally prefers forensic "bit-for-bit, byte-for-byte" duplicates of target systems. If you are responding to an incident that can evolve into a corporate-wide issue with grave consequences, you may want to perform a forensic duplication.

It is a good idea to have some policy that addresses when full duplication of a system is required. This may hinge on the system itself or the type of activity investigated. For example, you may choose to consider a sexual harassment suit or any investigation that can lead to the firing or demotion of an employee as grave enough to perform forensic duplication. If you are unsure, you can take the approach of imaging everything and sorting it out later.

SO WHAT?

Initial response is a stage of preliminary information gathering to determine whether or not unlawful, unauthorized, or unacceptable activity occurred. The information gathered during your initial response forms the basis for your level of response. During the initial response, it may be necessary to capture live data—volatile evidence—before it is lost through actions such as rebooting or shutting down a system. When live data collection is necessary, it is critical to adhere to sound forensic principles and alter the state of the system as little as possible. The information you obtain during the response may lead to administrative or legal proceedings.

The actions you take during the initial response are critical to foster good decisions later in an investigation. We have found that the best approach is an incremental one. Use the least intrusive commands first to determine the scope of the incident and to decide whether it warrants a full forensic duplication. In our opinion, if you have the resources and the technical capabilities, you can never go wrong with a full duplication.

QUESTIONS

1. On what media do you store and use your forensic toolkit? Why?
2. How do you determine which executables are associated with listening ports?
3. Why is it unnecessary to obtain application logs during live response?
4. Why is remotely viewing event logs not considered a sound practice?