

FOUNDSTONE
KNOW VULNERABILITIES

Initial Response to Windows NT/2000

By Kevin Mandia

Director of Computer Forensics

Foundstone, Inc.

Performing a Forensically Sound Initial Response to Windows NT/2000

There have been quite a number of NT/2000 systems getting attacked recently - and we have found there is a need for forensic and response procedures that adequately support any legal or administrative actions taken. We propose the following as a guideline for developing a sound methodology for NT Response.

Initial response is a stage of preliminary information gathering to determine whether or not unlawful, unauthorized, or unacceptable activity occurred on your networks. During the initial response, it is critical to capture volatile evidence before it is lost. It is also 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.

We consider three possible variations of initial response:

1. Obtaining merely the volatile data - performed when you know you will be making a forensic duplication.
2. Performing an In-Depth response to obtain enough information to determine if you need to perform a forensic duplication.
3. Performing a full investigation on a "live" system that you will not take offline for forensic duplication.

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 if 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. 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! Ask questions to those who are involved and knowledgeable about an incident.

The Common Mistakes

Failure to Document Findings Appropriately.

Failure to Notify or Provide Accurate Information to Decision Makers.

Failure to Record and Control Access to Digital Evidence.

Wait Too Long Before Reporting.

Installing Software on the Victim System (Why?)

Underestimating the amount and types of Evidence that may be found. (We rarely find an individual that is guilty of a single type of crime).

In this text, we outline the steps to take when performing the initial response to a Windows NT or Windows 2000 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 discuss how to gather live, volatile data that is critical to a complete investigation. We also provide an approach to an in-depth live recovery, where we explore obtaining as much information as needed from a live system to determine the who, what, when, where, and how of an incident.

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. Rule number one for initial response is don't destroy or alter the evidence. The best way to meet this goal is to take the time 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.

Toolkit Labels

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

Toolkit Contents

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. Instead, use only CUI or command-line tools during response on a Windows system. All of the tools discussed in this chapter are CUI or command-line 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 1.

Response Toolkit Tool	Description
cmd.exe	The command prompt for Windows NT and Windows 2000.
loggedon	A utility that shows all users connected locally and remotely.

Initial Response to Windows NT/2000

rasusers	An NTRK command that shows which users have remote-access privileges on the target system
netstat	A built-in system tool that enumerates all listening ports and all current connections to those ports.
fport	A utility that enumerates all processes that opened any TCP/IP ports on a Windows NT/2000 system
pstlist	A utility that enumerates all running processes on the target system
listdlls	A utility that lists all running processes, their command line arguments, and the Dynamically Linked Libraries each process depends on
nbtstat	A built-in system tool that lists the recent NetBIOS connections for approximately the last 10 minutes
arp	A built-in system tool that shows the MAC addresses of systems that the target system has been communicating with, within the last minute
kill	An NT Resource Kit (NTRK) command that terminates a process
md5sum	A utility that creates md5 hashes for a given file.
rmtshare	An NTRK command that displays the shares accessible on a remote machine.

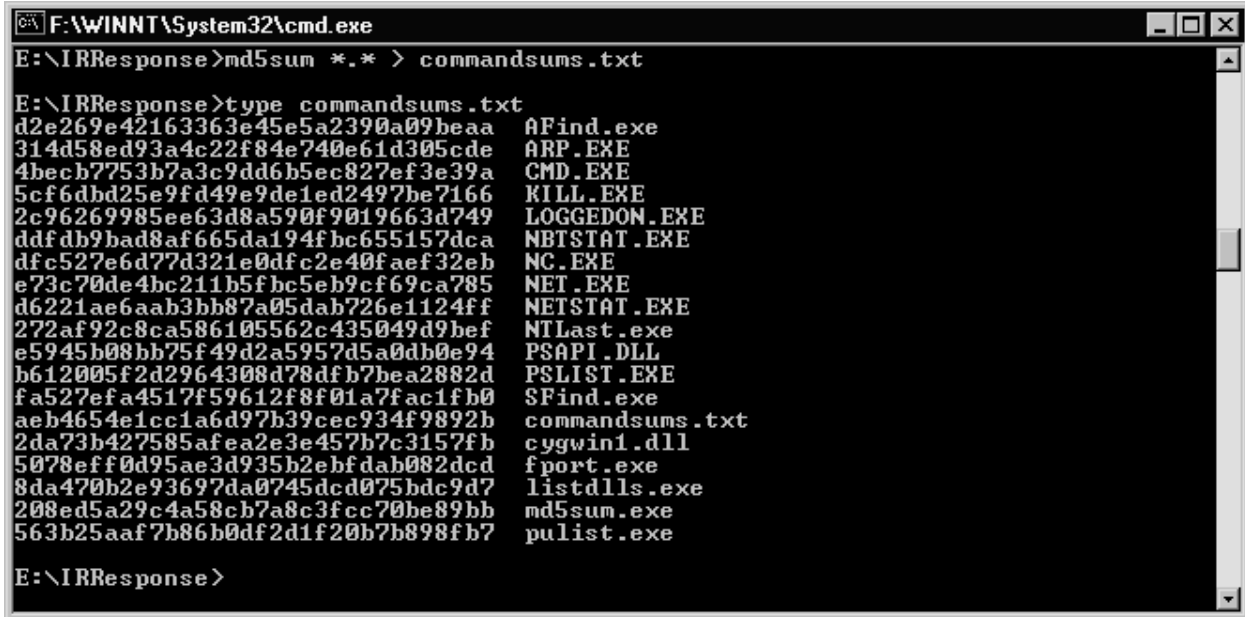
netctat (cryptcat)	A utility used to create a communication channel between two different systems. Cryptcat is used to create an encrypted channel of communications. Netcat provides a simple way to transfer information between networked systems.
doskey	A built in system tool that displays the command history for an open CMD.EXE shell.

Table 1: Response Toolkit Tools

<p>Where to get the tools:</p> <p>loggedon, pslist, listdlls, and filemon: www.sysinternals.com</p> <p>fport,: www.foundstone.com</p> <p>md5sum and cygwin.dll www.cygwin.com</p>

You need to ensure that your toolkit will function exactly as intended, and not alter the target system. Therefore, you will want to create a response disk that has all the dependencies (or as many as possible) covered. It is important to determine which dynamically linked libraries and files your response tools depend on. We recommend using filemon to determine all the files accessed and affected by each of these utilities. During the creation of the toolkit, we spend time performing filemon for each tool we use during response. It is good to know which tools change access times on files on the target system. We avoid using "loud" tools that alter a lot of the target system.

One of the files on our response kit floppy (or CD) is a text file with a checksum of all the commands on it. Figure 1 shows the md5sum command line used to create the text file (named commandsums.txt),



```

F:\WINNT\System32\cmd.exe
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  $Find.exe
aeb4654e1cc1a6d97b39cec934f9892b  commandsums.txt
2da73b427585afea2e3e457b7c3157fb  cygwin1.dll
5078eff0d95ae3d935b2ebfdab082dcd  fport.exe
8da470b2e93697da0745dcd075bdc9d7  listdlls.exe
200ed5a29c4a58cb7a8c3fcc70be89bb  md5sum.exe
563b25aaf7b86b0df2d1f20b7b898fb7  pulist.exe

E:\IRResponse>

```

Figure 1: Using md5sum to create a checksum for your response toolkit

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.

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 onto the response floppy disk or other removable media.
- Record the data you retrieve by hand in a notebook.
- Save the data you retrieve on the hard drive of the target system.

Initial Response to Windows NT/2000

- Save the data you retrieve on a remote "forensic system" using netcat or cryptcat (www.farm9.com).

We often choose netcat to transfer the information to a forensic workstation, and that approach is described in detail in this section. During crises with exigent circumstances, when you do not have the time to obtain a network connection, it is often easier to save the output files to a floppy disk. If the system has removable media, such as Iomega's Zip or Jaz drives, you may decide to store the information you retrieve there.

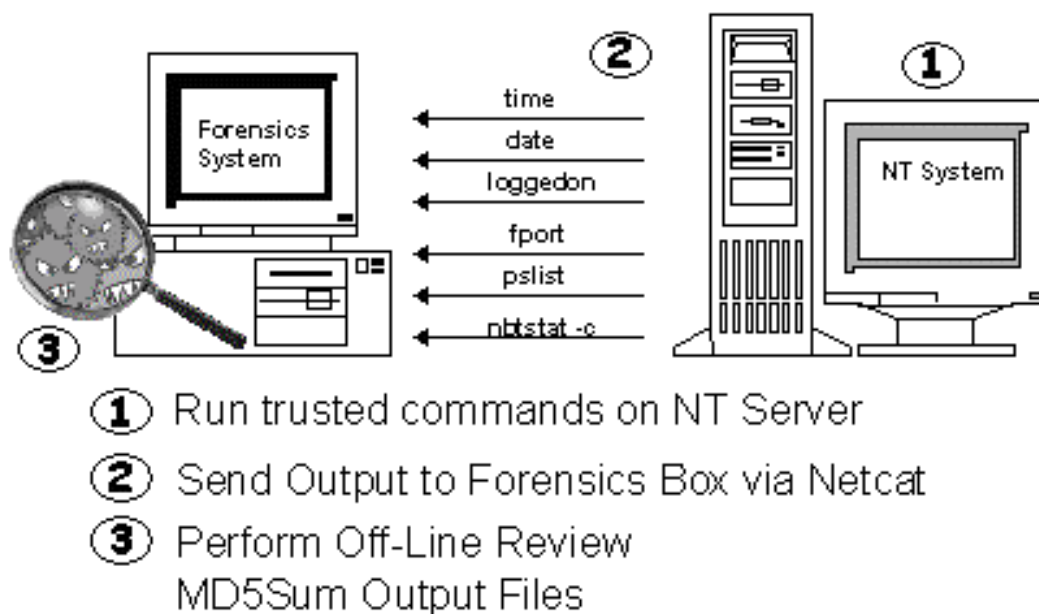
One of the most effective ways to retrieve information from target systems is to store the information on a remote forensic workstation using the tool netcat. All that you need to use netcat is an IP address on the target network and a laptop system with enough storage space to retain the information 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. The idea is to transfer the information via the target network, so that you can review it after you have executed your response. This technique of information gathering promotes two sound practices:

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

Netcat is a freely available tool that simply 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. Figure 2 illustrates the process of using netcat during initial response.

Using Netcat for Response



Done in an Organized, Forensically Sound Fashion

Figure 2: Using netcat during initial response to incidents

To use netcat, you initiate a *netcat listener* on the forensic workstation and redirect all incoming data to a file. Figure 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.

```

cmd.exe - nc -l -p 2222
E:\IRResponse>nc -l -p 2222 > pslist
-

```

Figure 3: Setting up the netcat listener on the forensic workstation

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

to a remote forensics workstation, you may need to break the connection by pressing CTRL-C on the forensic workstation. When the floppy or CD-ROM stops spinning on the target system, it is your cue to break the netcat listener connection on the forensics workstation.



Figure 4: Sending the output of pslist to the forensic workstation

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.

Cryptcat has the same syntax and function as the normal 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.

```
netcat http://www.l0pht.com/~weld/netcat/.
```

```
cryptcat: http://farm9.com/content/Free_Tools/Cryptcat
```

Obtaining Volatile Data Prior to Forensic Duplication

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, perform as few operations as possible to gather enough information to make the decision whether the incident warrants forensic duplication.

If you know that the incident you are investigating will require forensic duplication, then you will want to get the volatile data from the Windows NT/2000 system prior to turning off the system. Here is a list of some of the volatile data:

- System date and time

Initial Response to Windows NT/2000

- A list of currently running processes
- A list of currently open sockets
- The applications listening on open sockets
- A list of the users who are currently logged on
- A list of the systems that have current or had recent connections to the system

You want to sandwich all the commands you execute during a response with the time and date command. This is a forensically sound principle. If any time/date stamps changed outside the time frame you performed your response, then you are not accountable for creating such changes. You will also want to maintain a record of each command you executed. 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.

Notice that the steps for NT/2000 and Unix systems are the same:

- Establish a trusted shell.
- Record the system date and time.
- Determine who is logged on.
- Record open sockets.
- List Processes that open sockets.
- List currently running processes.
- List systems that recently connected.
- Record system time.
- Record the steps taken.

	NT/2000	Unix
	cmd.exe	<i>/bin/bash</i>
	date time	<i>w</i>
	loggedon	
	netstat	<i>netstat -anp</i>
	fport	<i>lsof</i>
	pslist	<i>ps</i>
	nbtstat	<i>netstat</i>
	date time	<i>w</i>
	doskey	<i>script, vi history</i>

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 the all the other employees. I have done most of my responses at night or on weekends, where the response is discrete. On the other hand, an active attack against your eCommerce 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. Have an md5sum file with the checksums of each tool you use prior to deployment. If you need to use untrusted binaries during a response, be sure to record the full path names of those binaries.

When responding at the console of a victim system, 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 md5sum 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	md5sum of output	Comments
12:15:22	type lmhosts nc 192.168.0.1 2222	X		3d2e.531d.6553.ee93.e089.0091.3857.eef3	Contents of lmhosts file
12:15:27	pslist nc 192.168.0.1 2222	X		1ded.672b.a8b2.ebf5.beef.6722.0100.3fe8	
12:15:32	netstat -an nc 192.168.0.1 2222	X		5228.5a23.1133.2453.efe2.0234.3857.eef3	

Using a form like this allows you to write down all the commands you are going to run before you respond on the target system. It forces the investigator to plan ahead!

It is a good idea to have a witness sign the form and verify each md5sum 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!

There are two reasons for diligently documenting your actions: to gather information that may become evidence against an individual and 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.

Executing a Trusted Cmd.exe

You always need to be careful of tripwires or booby traps that attackers put in place to foil incident response. You may run cmd.exe on a victim system and find out too late that del *.* was executed in the \WINNT\System32 directory, virtually making the system inoperable. The solution is to execute a trusted cmd.exe. Figure 5 illustrates using the Start | Run command on a Windows system to open a trusted cmd.exe on the floppy drive.



Figure 4: Running a trusted version of cmd.exe

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



```
MS-DOS A:\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:\>
```

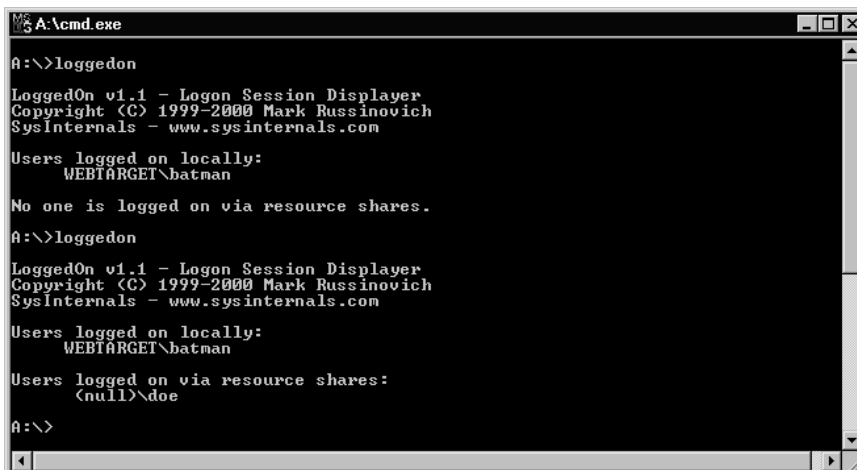
Figure 5: Obtaining the system time and date

Maintain a consistent naming convention for your output files. Also, as soon as you create a file, immediately generate an md5sum of the results. This helps to ensure the integrity of the document file.

Note: We often get all three time stamps - the modified, access, and creation times using the dir command as our second step. We feel much more comfortable responding on a system after we have already obtained the time/date stamps. However, if you know you are going to perform a forensic duplication, the time/date stamps ought to maintain integrity after gathering the volatile data from the kernel structures.

Determining Who Is Logged into the System

Mark Russinovich created loggedon, a utility that shows all users connected locally and remotely. Notice the null session connection from a remote system in Figure 7.



```
MS-DOS A:\cmd.exe
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 6: Using loggedon to list users currently logged into a system

Windows Remote Access Service

If you are responding to a system that allows dial-in modem access, you need to determine the user accounts that have remote-access privileges on the target system. If none do, then the modem is for outgoing connections (or at least not RAS). If several accounts can access the system via RAS, you need to decide whether or not you want to pull the phone 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 that can log into a system via RAS is rasusers.

Determining Open Ports and Listening Applications

Obviously, 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.

J.D. Glaser of Foundstone wrote a tool called fport, which enumerates all processes listening ports on a Windows NT/2000 system. Figure 8 shows the syntax for fport and the corresponding output.

```

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\inet_srv\inetinfo.exe
2 System -> 80 TCP
160 inetinfo -> 80 TCP D:\WINNT\System32\inet_srv\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\inet_srv\inetinfo.exe
2 System -> 465 TCP
160 inetinfo -> 465 TCP D:\WINNT\System32\inet_srv\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\inet_srv\inetinfo.exe
2 System -> 1031 TCP
160 inetinfo -> 1031 TCP D:\WINNT\System32\inet_srv\inetinfo.exe
2 System -> 1151 TCP
2 System -> 3970 TCP
160 inetinfo -> 3970 TCP D:\WINNT\System32\inet_srv\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 7: Using fport to view listening services

Common Backdoors and Their Default Ports

<http://www.doshelp.com/trojanports.htm>

<http://home.tiscalinet.be/bchicken/trojans/trojanpo.htm>

<http://www.simovits.com/nyheter9902.html>

Although fport shows the currently listening ports, it does not tell you which ports are currently servicing remote systems. For this information, 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 9 shows netstat being executed on an NT server.

```

DOS Prompt
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  D:\WINNT\System32\inetinfo.exe
125  inetinfo           -> 21   TCP  D:\WINNT\System32\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\inetinfo.exe
 2   System             -> 1027 TCP
125  inetinfo           -> 1027 TCP  D:\WINNT\System32\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:\invest>

```

Figure 9: Using netstat to view current connections and listening ports

You will notice there are 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.

If fport yields a rogue process listening for connections, and netstat shows current connections to that process, you may want to kill (terminate) the process to protect your system from potentially malicious actions taken by unauthorized intruders. When necessary, use the kill command to kill rogue processes.

Figure 10 shows the results when running fport on a system that has several remote-access trojans installed.

```

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        **:*
  
```

Figure 10: Recognizing unauthorized back doors

Process ID 162 does look suspicious, because \WINNT\winpop.exe is listening for connections on ports 6000 and 12346, which are ports commonly used by the Netbus trojan. Process ID 199 is also suspicious. The next step is to get both winpop.exe (the popular Netbus trojan) and windll.exe (the "girlfriend trojan") for further analysis. One quick solution is to copy both files to the response floppy, and then use an up-to-date virus scanner on another system to determine if these programs are remote-access trojans.

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 11 shows an example of running pslist.

```

A:\cmd.exe
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
GARDPWR  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
LLSSRU   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
NDDEAGMT 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
NTUDM    233 8   3  64   648     0:00:34.649 0:00:08.311 7:13:09.270
EUEMUIWR 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 11: Using pslist to view all running processes

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.

Initial Response to Windows NT/2000

If you cannot tell the difference between NT critical processes and rogue processes, then 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. Refer to Table 9-2 for a list of some NT/2000 system 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 2: Some Windows NT System Processes

Note: If you ever lose the desktop, for whatever reason (hung process), you can choose Start | Run and enter **Explorer**. The desktop should reappear.

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).

Listing Current and Recent Connections

Netstat, arp, and nbtstat are good utilities to use to determine who is connected or has recently connected to a system. Many NT/2000 workstations have audit policies that do not log any successful or failed logons. Therefore, these three utilities may be your only way to identify a remote system connecting to a workstation. Arp 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 is used to access the remote NetBIOS name cache, listing the recent NetBIOS connections for approximately the last ten minutes. Figure 12 shows an example of using nbtstat to list current and recent NetBIOS connections.

```

MS-DOS A:\cmd.exe
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 12: Using nbtstat to view recent NetBIOS connections

Note: Many computer security specialists use netstat to list the open ports on a system. Since nbtstat lists the open ports and the exact 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..

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



Figure 13: Using doskey to record the steps taken during incident response

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. Here is a sample script that can be used when responding to incidents on Windows NT/2000 systems:

```
time /t  
  
date /t  
  
loggedon  
  
netstat -an  
  
fport  
  
pslist  
  
nbtstat -c  
  
time /t  
  
date /t
```

Simply create a text file and add a .bat extension to it, and now you have a batch file. 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
```

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, in this case called iroutput.txt, with all the volatile information recorded in a neat fashion.

This concludes a simple response to record much of the volatile data from a Windows NT/2000 system. You may want to dump RAM, attain some information from the Registry, or perform numerous other actions on the target system, pending the totality of the circumstances. These steps merely establish the minimum baseline required to attain some critical data that is lost if you simply turn off the system and perform forensic duplication.

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

Initial Response to Windows NT/2000

exactly what to look for. Otherwise, you may be left with an incomplete response, without a proper purging of evidence or rogue processes and files.

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

- Run date and time to sandwich your response between a starting and ending time. This records the current system time for correlation between system logs and network-based logging.
- Use loggedon 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. 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.

The following sections outline an approach that obtains quite a bit of information from a live Windows NT/2000 system. These tools are presented in the order in which they are commonly used, but it is likely that you may need to alter the 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 conjunction with netcat to respond across a network connection.

Table

In Depth Response Toolkit Tool	Description
auditpol	An NTRK command line tool that determines the audit policy on a system.
reg	A NTRK command line tool used to dump specific information (keys) within the NT/2000

	Registry.
regdump	A NTRK command line tool that dumps the registry as a text file.
pwdump	A utility that dumps the SAM database so that the passwords can be cracked.
ntlast	A utility that monitors successful and failed logins to a system
sfind	A utility that detects files hidden within NTFS file streams.
afind	A utility that can search a file system to determine files accessed during specific time frames.
dumpel	A NTRK command line tool that is used to dump the NT/2000 event logs.

Table 3: Tools used for an In-depth response.

pwdump: <http://packetstorm.securify.com/Crackers/NT/pwdump2.zip>

afind, ntlast, sfind: <http://www.foundstone.com>

Obtaining Event Logs During Live Response

Use auditpol from the NTRK to query what 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 have been logged to the security log (event ID 612). Figure 14 shows the command line and output for auditpol.

```

A:\>auditpol
Running ...

<X> Audit Enabled

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

A:\>

```

Figure 14: Using auditpol to determine system logging

Ntlast, developed by Foundstone's J. D. Glaser, is an excellent tool that allows you to monitor successful and failed logins to a system, if the system's Logon and Logoff auditing was turned on. You will want to look for suspicious user accounts and remote systems accessing the target system. Figure 15 shows the successful logons to the system GENGIS, using ntlast.

```

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 15: Using ntlast to view successful logons

Use ntlast -r to list all successful logons from remote systems. Figure 9-16 shows an example of this form of ntlast.

Figure16: Using ntlast to list all successful logins from remote systems

```

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:\>_

```

Additionally, ntlast can be used to enumerate failed console logins using ntlast -f, as shown in Figure 9-17. To see failed remote logins, use ntlast -f -r.

```

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 17: Listing the failed logins at the system console

You will want to retrieve the other logs for offline analysis. Why search randomly on the target system using Event Viewer? Use dumpel and netcat to retrieve remote logs. Use dumpel -l security -t (in the NTRK) to obtain the event logs from the target system. This command dumps the entire security log, with tabs as the delimiter, to any file you specify. The dumpel -l application -t command dumps the application log to standard output.

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 Unicode attack. The trojan was placed in the directory where the default Web server scripts are stored. It is likely that the attacker had placed an Active Server Page that allowed her to upload arbitrary files.

```

3/4/01      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 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).



Figure 18: Connecting to a remote NT system 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 19.

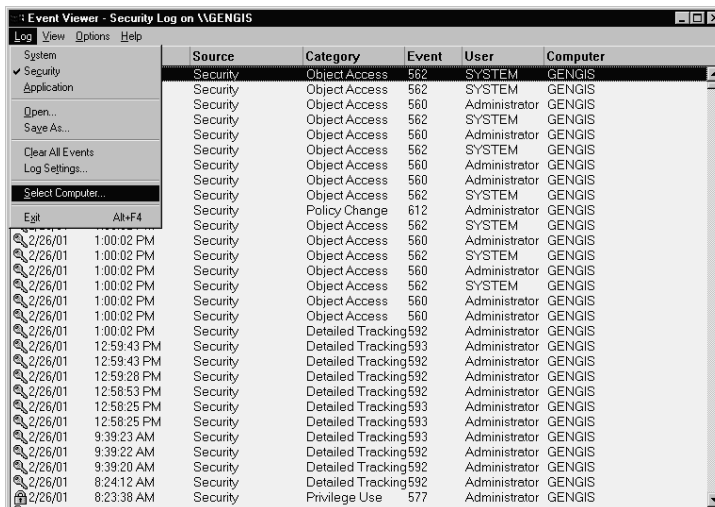


Figure 19: Using Event Viewer to review a remote system's event logs

Note: We included how to access the event logs via a network connection solely because we are frequently asked how remote administration of NT/2000 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 NT/2000 Registry stores a wealth of important data that is useful during initial response. We cover the full details of investigating the Registry in Chapter 10.

For live retrieval of the important Registry data, you can use regdump or reg query, both from the NTRK. Regdump creates an enormous text file of the Registry. We use reg query

Initial Response to Windows NT/2000

instead and extract just the Registry key values of interest. Here is a sample batch file that we have used to get some information off of 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"
```

Initial Response to Windows NT/2000

```
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.

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 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!* Although time/date stamps are volatile data - if you know you are performing a forensic duplication, the time/date stamps will be maintained after you obtain the volatile data and shut the system down.

If you can identify the relevant time frame when an incident occurred, the time/date stamps become the evidence of which files an attacker touched, uploaded, downloaded, and executed. Although this takes a long time on UNIX systems, Windows performs this task

extremely quickly. Here are examples of using dir to obtain access, modification, and access 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

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 pwdump by Todd Sabin to dump the passwords from the Security Access Manager (SAM) database. These passwords may be cracked on a forensic workstation using John the Ripper, L0phtcrack, or any other NT password-cracking tool. Remember, 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.

Dumping System RAM

It may be important for you to dump the contents of memory—perhaps to obtain passwords, get the clear text of a recently typed encrypted message, or retrieve the contents of a recently opened file. Unfortunately, Windows NT/2000 support for memory dumping does not correspond with sound forensic procedures.

There are two ways to dump the contents of memory in NT: through the GUI or via editing the Registry. If you choose to edit the Registry, then you must do so based on the file system you currently have (either FAT or NTFS). The memory dump process creates a file on the target hard drive, unless you can use a network drive on a remote system. Either way, you need to reboot the system. Therefore, if you feel a memory dump is critical to your investigation, you might as well plan on performing a forensic duplication of the system.

To Attain Information on NT Memory Dumps

<http://support.microsoft.com/support/kb/articles/Q235/4/96.ASP>

To Attain Information on Windows 2000 Memory Dumps:

<http://support.microsoft.com/support/kb/articles/Q254/6/49.ASP>

L0phtcrack: <http://www.securitysoftwaretech.com/l0phtcrack/>

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

Is Forensic Duplication Necessary?

After reviewing the system information you retrieved during the initial response, you need to decide whether or not to perform a forensic duplication of the evidence. Generally, if the incident is severe or deleted material may need to be recovered, a forensic duplication is warranted. The forensic duplication of the target media provides the "mirror image" of the target system, which shows due diligence when handling critical incidents. It also provides a means to have working copies of the target media for analysis without worrying about altering or destroying potential evidence.

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.