GIAC Security Essentials

Practical Assignment Version 1.4b
Online

Submitted by: Tan Koon Yaw

Table of Content

Windows Responder's Guide

Abstract

When a system encounters an incident, there is a need to handle the case properly to gather evidence and investigate the cause. Initial response is the stage where preliminary information is gathered to determine whether there is any breach of security and the possible causes if any. This paper provides the first responder guide to handle incident occur on a Windows platform system.

In this paper, we will discuss what are the issues one needs to consider during the initial response stage. There are critical evidence that need to be protected and gathered during the initial response stage. We will hence discuss what are the tools that can be used to gather the necessary evidence and how to collect them appropriately. Finally, we will explore areas that one needs to look out for during the investigation on the evidence collected.

1. Introduction

When a system encounters an incident, the common reaction among most people will be to panic and jump straight into the system to find out the cause and hopefully try to get it back to normal working condition as soon as possible. Such knee-jerk reactions is especially so for systems supporting critical business operations. However, such actions may tamper with the evidence and even lead to a lost of information causing potential implications. This is especially critical if the recourse actions involve legal proceedings. Hence it is very important to establish a set of proper and systematic procedures to preserve all evidence during this critical initial response stage.

Not every incident will lead to a full investigation or legal proceeding. However, in the event when a security breach has taken place, proper handling of the system is necessary. However, one should always bear in mind that different incidents might require different procedures to resolve.

In most cases, not all systems can afford the downtime to carry a full investigation before knowing the most possible cause. Initial response is the stage of preliminary information gathering to determine the probable causes and the next appropriate response. Responders should be equipped with the right knowledge on how and what information to collect without disrupting the services. During the initial response, it is also critical to capture the volatile evidence on the live system before they are lost.

This paper will cover the initial response focusing on the windows platform, how and what evidence should be collected and analyzed quickly. We will begin the discussion on what is initial response, what are the potential issues need to be

considered, what to do and what not to do during the stage of initial response. To carry out the initial response successfully, the responder needs to prepare a set of tools to gather the evidence. We will list out some of the essential tools that a responder should be equipped and run through how and what evidence should be collected. This paper will not cover the forensic investigative analysis process. However, areas to look out for footprints of intrusion on the system will be discussed.

2. Initial Response

Initial response is the stage where preliminary information is gathered to determine whether there is any breach of security, and if so, to determine the possible breach and assess the potential impact. This will allow one to determine what is the next course of action, whether to let the system continues its operation or arrange for immediate isolation for a full investigation.

During the initial response stage, the following questions (Who, What, When, Where, How) should be asked:

- Who found the incident?
- How was the incident discovered?
- When did the incident occur?
- What was the level of damage?
- Where was the attack initiated?
- What techniques were being used to compromise the system?

There should be a well-documented policy and procedures on how different types of incidents should be handled. It is also important to understand the policies and response posture. The level of success to solve an incident does not depend only on the ability to uncover evidence from the system but also the ability to follow proper methodology during the incident response and evidence gathering stage.

When one suspects a system is compromised, the natural question is to ask whether to bring the system offline, power off the system or let it remains. For a compromised system, do you intend to collect evidence and trace the attacker or just patch the system and life goes on? There is no right answer to this. It really depends on the organization business needs and response plan. For example, when one suspects the attacker is still on the system, you may not want to alert him/her by pulling the system offline immediately, but let the system remains and continue to monitor the his/her activities before taking appropriate actions. However, for system that contains sensitive information, there may be a need to pull the system offline immediately before incurring further damage.

3. Evidence Gathering

Electronic media is easily manipulated, thus a responder needs to be careful when handling evidence. The basic principles to keep in mind when gathering the evidence is to perform as little operations on the system as possible and maintain a detailed documentation on every single steps on what have been done to the system.

Majority of the security incidents do not lead to civil or criminal proceedings. However, it is to the best interest of the organization to treat the incidents with the mindset that every action you take during incident response may later lead to legal proceeding or one day under the scrutiny of individuals who desire to discredit your techniques, testimony or basic finding skills.

Maintaining a chain of custody is important. Chain of custody establishes a record of who handle the evidence, how the evidence is handled and the integrity of the evidence is maintained.

When you begin to collect the evidence, record what you have done and the general findings in a notebook together with the date and time. Use a tape recorder if necessary. Note that the system that you are working on could be rootkited.

Keep in mind that there are things to avoid doing on the system:

- Writing to the original media
- Killing any processes
- Meddling the timestamp
- Using untrusted tools
- Meddling the system (reboot, patch, update, reconfigure the system).

4. Protecting the Volatile Information

When the system is required to undergo the computer forensic process, it is necessary to shutdown the system in order to make bit-level image of the drive. There are discussions on how system should be shutdown, and we are not going to cover this in details here. However, by shutting down the system, a great deal of information will be lost. These are the volatile information, which include the running processes, network connection and memory content. It is therefore essential to capture the volatile information on the live system before they are lost.

The order of volatility is as follows:

- Registers, cache contents
- Memory contents
- State of network connections
- State of running processes
- Contents of file system and hard drives
- Contents of removable and backup media

For the first four content, the information are lost or modified if the system is shutdown or rebooted.

Some of the volatile evidence that are important to gather are:

- System date and time
- Current running and active processes
- Current network connections
- Current open ports
- Applications that listening on the open sockets
- Current logon users

Such volatile evidence is important, as it will provide the critical first hand information, which may make or break a case. In some cases, some hackers may have tools that run in memory. Gathering such evidence is therefore necessary as part of the initial response procedure.

5. Creating a Response Toolkit

Preserving evidence and ensuring those evidence that you gather is correct is very important. There is a need to ensure the programs and tools that one uses to collect the evidence are trusted. Burning them into a CD-ROM media will be ideal to carry them around when responding to incidents. The responder should always be equipped with the necessary programs beforehand. This will shorten the response time and enable a more successful response effort.

There are many tools available that can be used to gather evidence from the system. Below is a list of tools that you should minimally be equipped with. There could be more depending how much you wish to carry out prior to bit-level imaging of the media. The important is to harvest the volatile information first. Those residing on the media could still be retrieved during the forensic analysis on the media image.

You need to ensure the tools that you used will not alter any data or timestamp of files in the system. It is therefore important to create a response disk that has all

the dependencies covered. The utility, filemon, could be used to determine the files being accessed and affected by each of the tool used.

Below is the set of response tools you should prepare:

Tools	Description	Where to get
cmd.exe	Command prompt.	From a trusted system
ipconfig	A system tool that enumerates IP address of the system.	From a trusted system
netstat	A system tool that enumerates listening ports and network connections.	From a trusted system
nbtstat	A system tool that displays NetBIOS over TCP/IP (NetBT) protocol statistics, NetBIOS name tables for both the local computer and remote computers, and the NetBIOS name cache.	From a trusted system
date	A system tool that enumerates the system date.	From a trusted system
time	A system tool that enumerates the system time.	From a trusted system
env	A utility that enumerates system variables.	http://unxutils.sourceforge.net/
psuptime	A utility that tells you how long a Win NT/2K system has been up.	http://www.sysinternals.com/ntw2k/freeware/psuptime.shtml
net	A system tool that enumerates NetBIOS connections, user accounts, share folders, start services etc.	From a trusted system
psloggedon	A utility that shows all users connected locally and remotely.	http://www.sysinternals.com/ntw2k/freeware/psloggedon.shtml
pulist	A command-line tool that displays active processes running on local or remote computers. It also captures the user running the processes.	http://www.microsoft.com/windows2000/techinfo/reskit/tools/existing/pulist-o.asp
pslist	A command-line tool shows CPU-oriented information for all the processes that are currently running on the local system. The information listed for each process includes the time the process has executed, the amount of time the process has executed in kernel and user modes, and the amount of physical memory that the OS has assigned the process. Command-line switches allow you to view memory-oriented process information, thread statistics, or all three types of data.	http://www.sysinternals.com/ntw2k/freeware/pslist.shtml
listdlls	A utility that list all the DLLs that are currently loaded, including where they are loaded and their version numbers.	http://www.sysinternals.com/ntw2k/freeware/listdlls.shtml
fport	A utility that identify open ports and their associated applications.	http://www.foundstone.com/resources/intrusion_detection.htm

psservice	A utility that displays the status, configuration, and dependencies of a service, and allows you to start, stop, pause, resume and restart them.	http://www.sysinternals.com/ntw2k/freeware/psservice.shtml
psinfo	A command-line tool that gathers key information about the local or remote Windows NT/2000 system, including the type of installation, kernel build, registered organization and owner, number of processors and their type, amount of physical memory, the install date of the system, and if it's a trial version, the expiration date.	http://www.sysinternals.com/ntw2k/freeware/psinfo.shtml
arp	A system tool that maps the logical IP address to physical MAC address.	From a trusted system
hfind	A utility that find files that have hidden attribute set.	http://www.foundstone.com/resources/proddesc/forensic-toolkit.htm
streams	A utility to view NTFS file stream information.	http://www.sysinternals.com/ntw2k/source/misc.shtml
ntlast	A utility that monitors successful and failed login to the system.	http://www.foundstone.com/resources/proddesc/ntlast.htm
reg	A command-line registry manipulation. Allow you to query the registry entries.	From trusted NT Resource Kit
auditpol	A command-line tool that determines the audit policy on a system.	From trusted NT Resource Kit
regdmp	A command-line tool that dumps the registry as a text file.	From trusted NT Resource Kit
md5sum	A utility that generated a hash value of a file.	http://unxutils.sourceforge.net/
netcat (cryptcat)	A utility that reads and writes data across network connections. Cryptcat is an equivalent version of netcat but create an encrypted channel of communication.	http://www.atstake.com/research/tools/network_utilities/
cat	A utility that is the equivalent of cat in the Unix world.	http://unxutils.sourceforge.net/
find	A utility that is the equivalent of find in the Unix world.	http://unxutils.sourceforge.net/
grep	A utility that is the equivalent of grep in the Unix world.	http://unxutils.sourceforge.net/
filemon	A utility that monitors and displays file system activity on a system in real-time. It allows one to explore the way Windows works, seeing how applications use the files and DLLs.	http://www.sysinternals.com/ntw2k/source/filemon.shtml
pclip	A utlity that put the Windows clipboard text to stdout.	http://unxutils.sourceforge.net/
tcpdump windump	A tool for network sniffer/analyzer. Windump is the Windows platform for tcpdump.	http://www.tcpdump.org/ http://windump.polito.it/

6. Gathering the Evidence

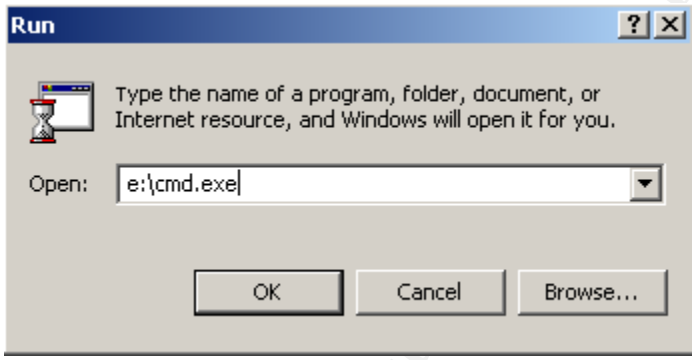
A critical question to ask someone when you encounter a live system is whether the system has been rebooted. It will be great news if the answer is no, but a yes reply is usually not a surprise.

Albeit the system has been rebooted and caused some vital information to be lost, it is still a good practice to carry out the initial response steps to gather the evidence prior to shutting down the system, as you will never know there could still be some other footprints around.

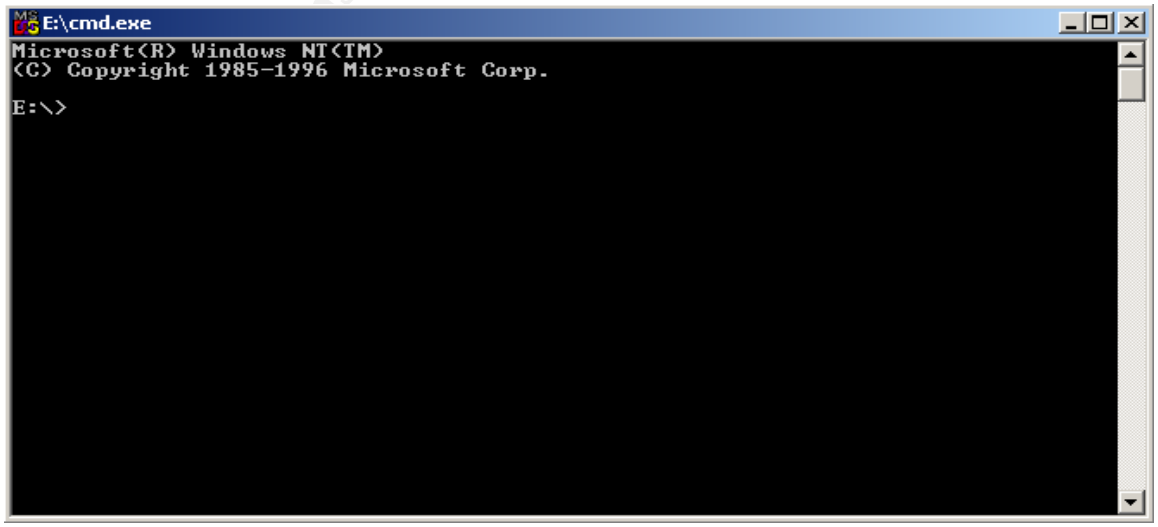
Step One: Open a Trusted Command Shell

The first step is to ensure all the tools are run from a trusted command shell.

Initiate a command shell from the Start Menu. Run the trusted command prompt from the trusted tools from the CD you have prepared.



All subsequent commands should then be run over this trusted shell.



Step Two: Prepare the Collection System

Remember that you should not write the evidence collected to the original media. A simple way is to write the data to a floppy disk. However, some of the evidence collected may exceed the disk space of the floppy disk. One simple way is to pipe the data over the network to your responder's system. To do this, we could use the popular known "TCP/IP Swiss Army Knife" tool, netcat, to perform the job.

The process of setting up the netcat is first by setting up the netcat listener on your responder's system.

```
D:\>nc -l -p 55555 >> evidence.txt
```

The above command open a listening port on your responder's system and redirect anything received to evidence.txt. The switch `-l` indicates listening mode. The listener will close the socket when it receives data. To allow the listener to continue to listen harder after the first data is captured, use the `-L` switch instead. Thus, you can choose whether to create a new file for each command or appending all evidence gathered into one single file by using the appropriate switch. The switch `-p` allows you to select the port for the listener. You could choose any other port.

When the listener is ready, you can start to pipe the evidence to the responder's system by executing the following (assuming E Drive is the CD ROM Drive):

```
E:\>nc <IP address of responder's system> <port> -e <command>
```

OR

```
E:\<command> | nc <IP address of responder's system> <port>
```

For example, if you want to pipe the directory listing to the responder's system (with IP address 10.1.2.3), you execute:

```
E:\> nc 10.1.2.3 55555 -e dir
```

OR

```
E:\dir | nc 10.1.2.3 55555
```

Note that the evidence pipes through netcat is in clear. If you prefer to encrypt the channel (for example, you suspect there is a sniffer on the network), you can use cryptcat. Cryptcat is the standard netcat enhanced with twofish encryption. It is used in the same way as netcat. Note that the secret is hardcoded to be "metallica" (use the `-k` option to change this key).

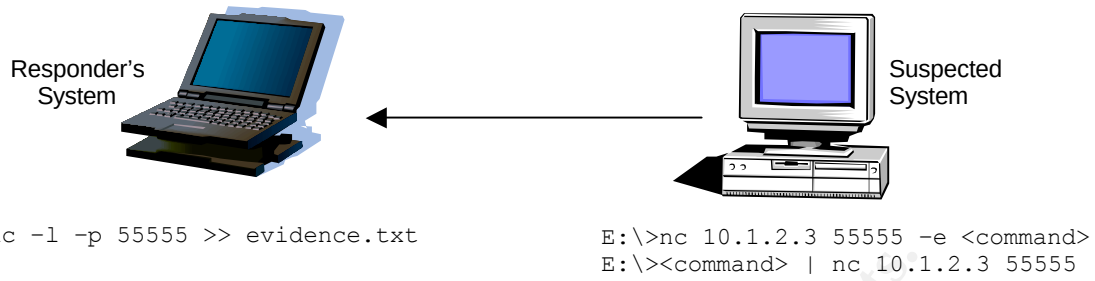


Figure 1: Using netcat to collect evidence

Step Three: Collect Volatile Evidence

Now you can start running your toolkit to collect the volatile evidence.

The necessary evidence to collect is:

- Basic system information
- Running processes
- Open sockets
- Network connections
- Network shares
- Network users

The system date and time should be recorded before and after collecting the evidence.

Command	Purpose
`date /t` `time /t`	To gather the date and time of the system. The system date and time must be checked with the current date and time to see any deviation. This is important when there is a need to correlate with logs from different sources, for example the firewall and IDS logs.
`ipconfig /all`	To gather the IP address configuration of the system.
`env`	Record the system environment. To understand the environment and identify any unusual environment being set. See Appendix, Figure A-1 for a screenshot on env listing.
`psinfo`	Record the system information, including the hotfixes applied.

	See Appendix, Figure A-2 for a screenshot on psinfo listing.
`psuptime`	Record the uptime of the system. See Appendix, Figure A-3 for a screenshot on psuptime listing.
`psloggedon`	Record users connected locally and remotely. To identify any unusual users on the system.

`ntlast –r`
`ntlast –f`
`ntlast –r -f`
	To record the successful and failed logins to the system.

`net use`
`net session`
`net file`
`net share`
`net view`
`net user`
`net accounts`
`net localgroup`
`net start`
	To gather the system information. This includes what are the share folders, user accounts, local group, start services etc.
	
	To identify unusual connections, users and services.
	
	See Appendix, Figure A-4 for a screenshot on net start listing.

`nbtstat -n`
`nbtstat –c`
`nbtstat -s`
	To access the remote NetBIOS name cache, listening the recent NetBIOS connections for approximately the last ten minutes.
	
	To gather the local NETBIOS name and NetBIOS Table cache of remote systems.
	
	To identify unusual logon users.

`pclip`	Retrieve content of the Clipboard. To identify any unusual content on the clipboard.

`pslist`
`pulist`
`psservice`
	Gather process information.
	
	pslist to gather running processes.
	pulist to gather active processes.
	
	To identify unusual processes.
	
	See Appendix A, Figure A-5 to A-7 on a screenshot on the various commands listing.

`listdlls`	Record all the DLLs that are currently loaded, including where they are loaded and their version numbers.

	To identify unusual DLLs, open files and Trojans. See Appendix A, Figure A-8 on a screenshot on listdlls listing.
`fport`	Record opening ports. To identify opening ports and their applications. Useful checklist: http://www.neohapsis.com/neolabs/neo-ports/neo-ports.svcs http://www.simovits.com/nyheter9902.html http://isc.sans.org/port_details.html See Appendix A, Figure A-9 on a screenshot on fport listing.
`netstat -an`	Record listening services. Record open ports. To identify unusual ports and connections.
`netstat -rn`	Record routing table. To identify unusual network connections
`arp -a`	Maps logical IP address to Physical MAC address.

`dir /t:a /a /s /o:d c:`
`dir /t:w /a /s /o:d c:`
`dir /t:c /a /s /o:d c:`
	Gather Last Access Time.
	Gather Last Modification Time
	Gather Last Create Time
	
	Only the files on the C Drive are recorded. For other drives, change the drive label according.
	
	To correlate timestamp of files with the unusual files, processes and connections.
	
	See Appendix A, Figure A-10 to A-12 on a screenshot on the various listings.

`hfind c:`	Record files in C Drive that have hidden attribute set. To identify suspicious files with hidden attribute set. See Appendix A, Figure A-13 on a screenshot on hfind listing.
`md5sum <file>`	Create the md5 checksum of all the files collected to ensure the information is not manipulated subsequently.

Some of the evidence gathered may seem normal but when all the evidence are collected, they provide a good picture of the system. From there, one can trace the normal and unusual processes, connections and files occurring in the system.

Step Four: Collect Pertinent Logs

After gather the volatile information, the next thing is to gather the pertinent logs. While this information is not considered to be volatile and could be retrieved during the forensic investigation, getting these information will still be helpful to get the first hand knowledge of the cause. Note that bit-level image of the media could take a while and during this period, investigation can be started on these logs first.

The pertinent logs to gather are:

- Registry
- Events logs
- Relevant application logs

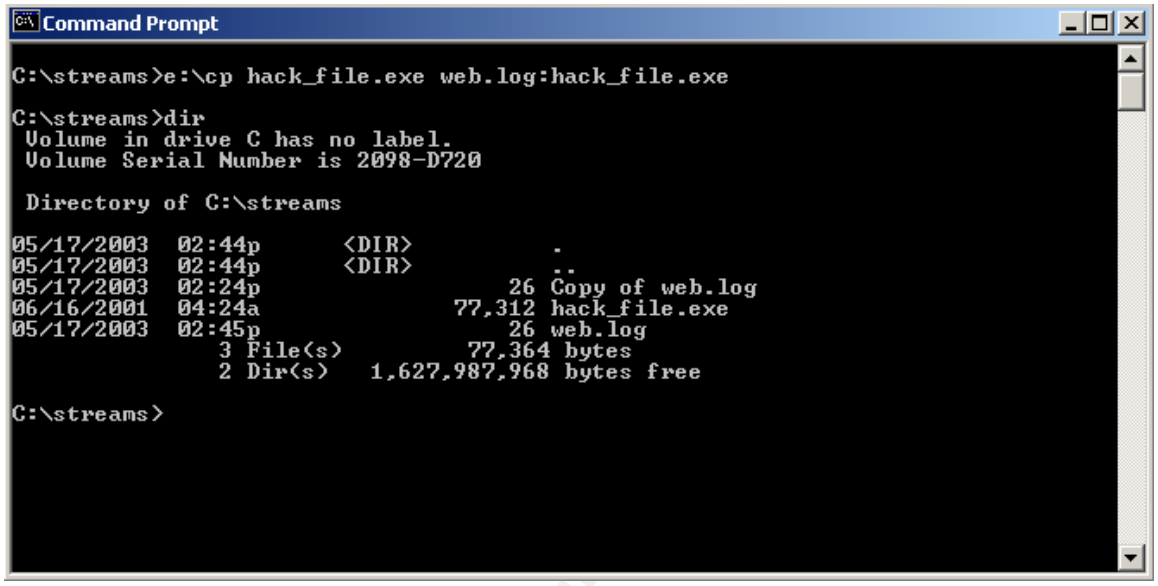
Command	Purpose
`auditpol`	To determine what policies exists on the system.

`reg query HKLM\Software\Microsoft\Windows\CurrentVersion\Run /s`
 	
`reg query HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce /s`
 	
`reg query HKLM\Software\Microsoft\Windows\CurrentVersion\RunServices /s`
 	
`reg query HKLM\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce /s`
	Retrieve the content of all “Run”, “RunOnce”, “RunServices” and “RunServiceOnce” keys and all subkeys.
	
	To identify unusual programs and Trojans.

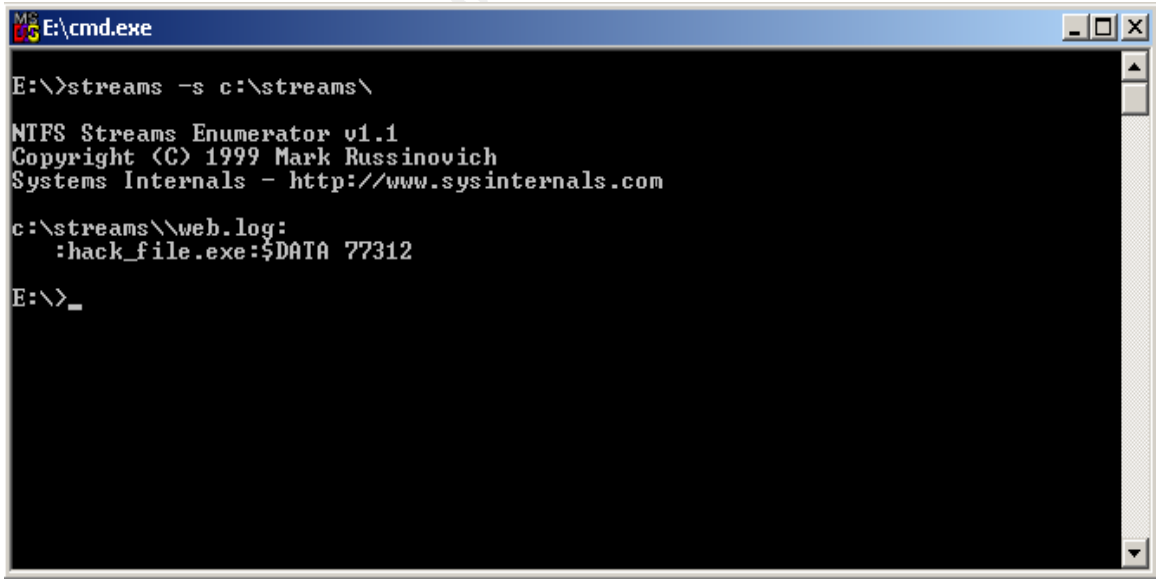
`reg query HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\`	Review Most Recently Used (MRU) files. To identify unusual files.
`reg query HKLM\Software\Microsoft\Windows\CurrentVersion\Uninstall`	Trace improper uninstall of programs. To identify unusual programs and Trojans.
`streams –s c:`	Check any NTFS streams on the C Drive.

Note that an attacker can make use of the NTFS stream to hide files. For example, the following will allow the attacker to hide the file, hack_file.exe, in web.log.

```
C:\> cp hack_file.exe web.log:hack_file.exe
```

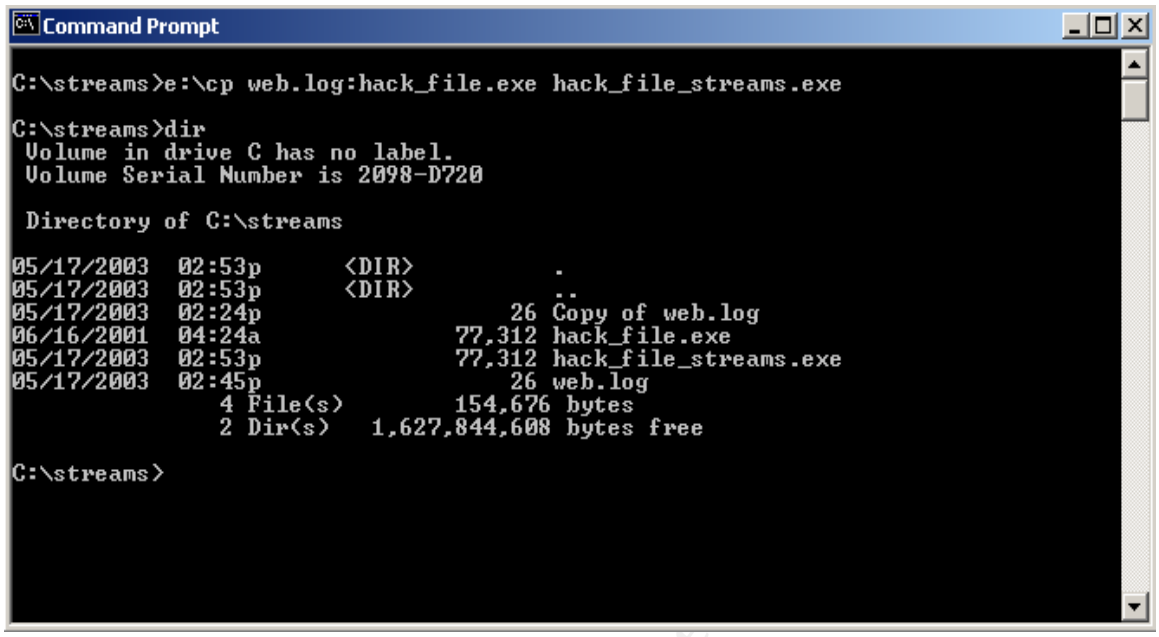


The file size of web.log will not change. To identify stream file, use the streams command.



To obtain the stream file, you just need to reverse the process:

```
C:\> cp web.log:hack_file.exe hack_file.exe
```



Stream file can be executed by START command:

```
C:\> start web.log:hack_file.exe
```

Event logs and other application logs are next to collect. They could be piped over to the responder's system using the `cat` utility. The default locations are as follows:

Logs	Default Location

Event logs	`c:\winnt\system32\config\AppEvent.Evt`
	`c:\winnt\system32\config\SecEvent.Evt`
	`c:\winnt\system32\config\SysEvent.Evt`

IIS Web logs (if any)	`c:\winnt\system32\logfiles\W3SVC1\`
Mail logs (if any)	`c:\winnt\system32\logfiles\SMTPSVC1\`

After the files are captured into the responder's system, you should make a md5sum on the files to ensure the integrity of the files are not tampered when carry out subsequent investigation.

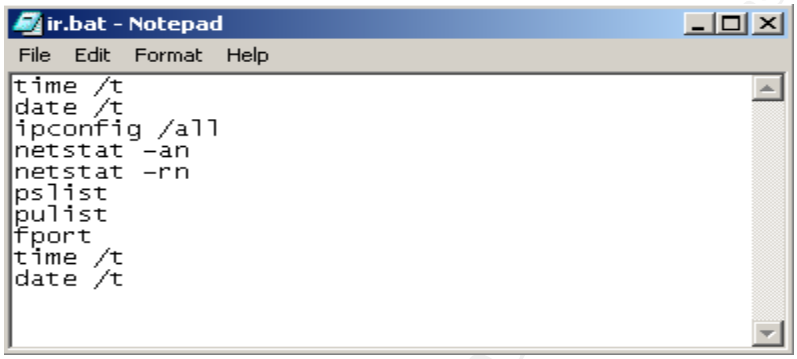
Step Five: Perform additional network surveillance

Where possible, it is good to monitor closely any connection to the system subsequently, especially if you suspect the attacker might return. Running a sniffer program on another system to monitor the network activities on that suspected system would be good.

Tool	Purpose
tcpdump/windump/ethereal/snort	Network Monitoring. Information gathered from netstat. Did the attacker come back?

7. Scripting the Initial Response

The commands used to gather the evidence can be written in a batch file. This will make the job of the responder easier and at the same time avoid mistyping the command. A simple way to create a script is to create a text file and give a .bat extension to it. This will give us a very neat way to collect evidence from the system. For example, we could key in the following as a single text file with file name ir.bat:



8. Identification of Footprints

You have now collected:

- Basic system information
- Running processes
- Open sockets
- Network connections
- Network shares
- Network users
- Pertinent logs

The next step is to identify the footprints. During the review, one should look out for the following:

- Check for hidden or unusual files
- Check for unusual processes and open sockets
- Check for unusual application requests
- Examine any jobs running

- Analyze trust relationship
- Check for suspicious accounts
- Determine the patch level of the system

Whenever there is any suspicious observation, take note of the event and timestamp. Correlate the event with other logs based on related files, processes, relationship, keywords and timestamp. The timestamp will also be useful to correlate with external logs such as the logs from firewall and intrusion detection system. Any suspected events should not be left out.

If one is analyzing IIS records, note that it uses UTC time. This is supposed to help to synchronize when running servers in multiple time zones. Windows calculates UTC time by offsetting the value of the system clock with the system time zone. Take note of this when you correlate the entries of the IIS logs with timestamp of other logs.

The Registry provides a good audit trail:

- Find software installed in the past
- Determine security posture of the machine
- Determine DLL Trojan and startup programs
- Determine Most Recently Used (MRU) Files information

9. What's Next?

Based on initial response finding, one should be able to determine the possible cause of the security breach and decide the next course of action whether to:

- Perform a full bit-level imaging for full investigation;
- Call the law enforcer; or
- Get the system back to normal (reinstall, patch and harden the system).

For bit-level disk image, there are tools out that that could perform an excellent job. Encase and SafeBack are two of the commercial tools that you could consider for image acquisition and restoration, data extraction, and computer forensic analysis. Another tool that you can consider is dd, which is free. dd is a utility that comes with most Unix platform. Now it has ported to Windows platform as well and you can get it at http://unxutils.sourceforge.net/.

10. Wrapping Up

In the event of any incident, having a proper initial response plan and procedure is important to ensure the evidence gathered is intact and at the same time do not tamper the evidence as far as possible. Volatile information is critical to

protect and ensure they are collected first before they are lost. Sometimes such information may make or break a case.

By having a good preparation to response to any security incidents will save a lot of time and effort in handling cases. Planning ahead is necessary for initial response. Never rush to handle an incident without any preparation.

Having said all these, the next step after a good preparation is practice. The actions taken during the stage of initial response is critical. Do not wait for an incident to occur before you start to kick in your established plan, checklist and toolkit. Remember practice makes perfect.

References

H. Carvey, “Win2K First Responder’s Guide”, 5 September 2002, URL: http://www.securityfocus.com/infocus/1624

Jamie Morris, “Forensics on the Windows Platform, Part One”, 28 January 2003, URL: http://www.securityfocus.com/infocus/1661

Stephen Barish, “Windows Forensics: A Case Study, Part One”, 31 December 2002, URL: http://www.securityfocus.com/infocus/1653

Stephen Barish, “Windows Forensics - A Case Study: Part Two”, 5 March 2003, URL: http://www.securityfocus.com/infocus/1672

Mark Burnett, “Maintaining Credible IIS Log Files”, 13 November 2002, URL: http://www.securityfocus.com/infocus/1639

Norman Haase, “Computer Forensics: Introduction to Incident Response and Investigation of Windows NT/2000”, 4 December 2001, URL: http://www.sans.org/rr/incident/comp_forensics3.php

Lori Willer, “Computer Forensics”, 4 May 2001, URL: http://www.sans.org/rr/incident/comp_forensics2.php

Kelvin Mandia and Chris Prosise, “Incident Response: Investigating Computer Crime”, Osborne/McGraw-Hill, July 2001, ISBN: 0-07-213182-9

http://unxutils.sourceforge.net/

http://www.sysinternals.com/

http://www.foundstone.com/

http://www.atstake.com/

http://www.guidancesoftware.com/

http://www.forensics-intl.com/

Appendix A

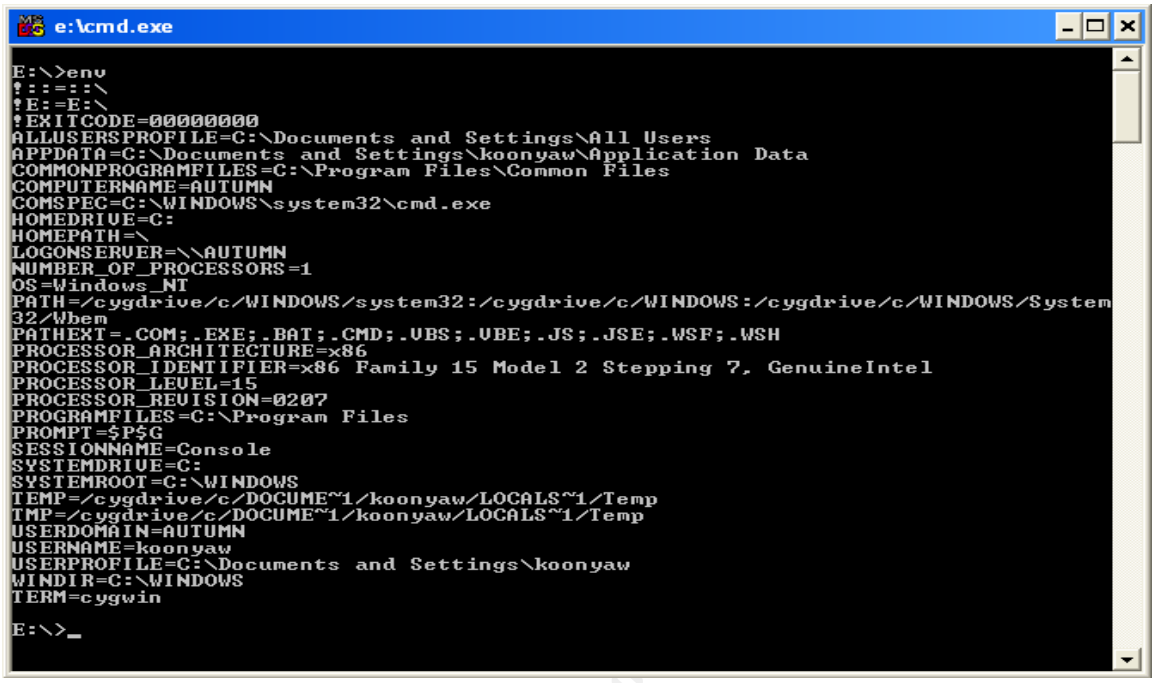


Figure A-1: env

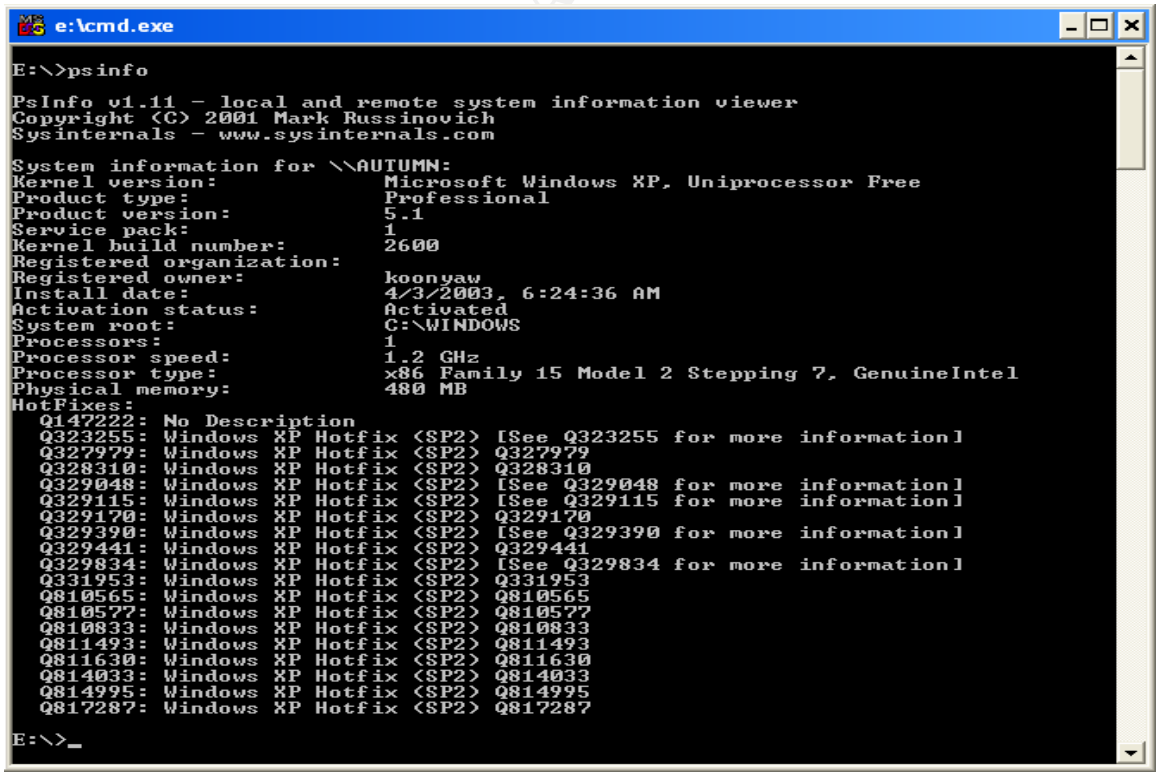


Figure A-2: psinfo

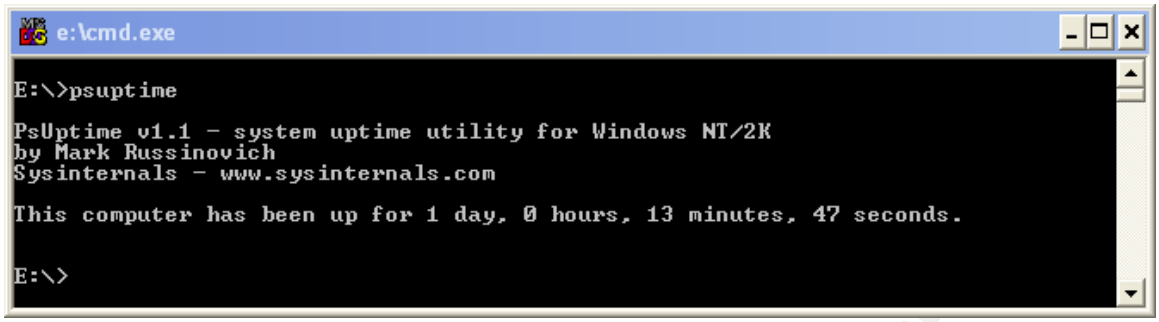


Figure A-3: psuptime

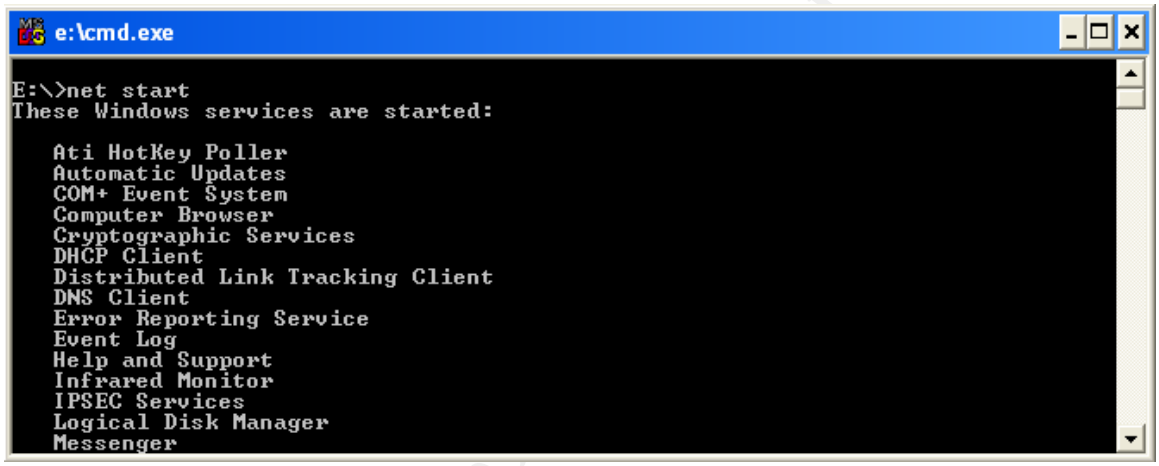


Figure A-4: net start

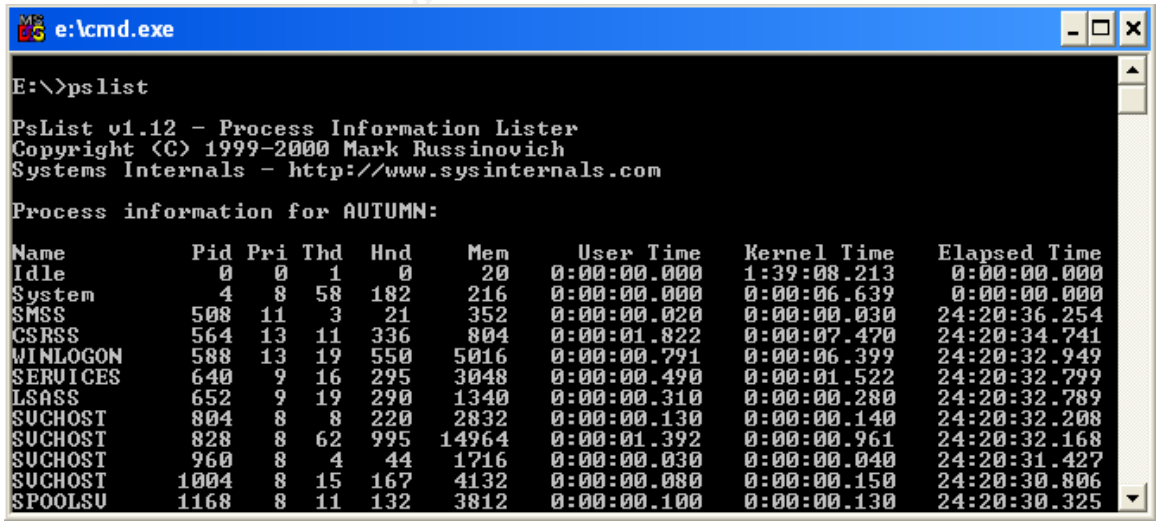


Figure A-5: pslist

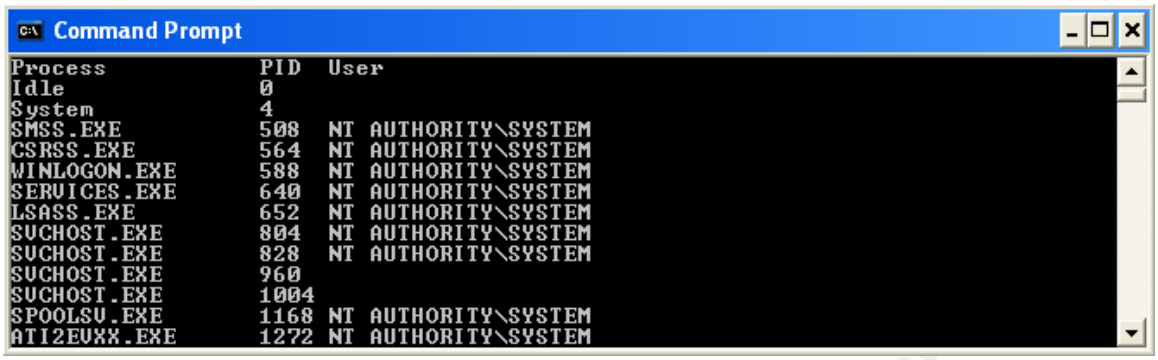


Figure A-6: pulist

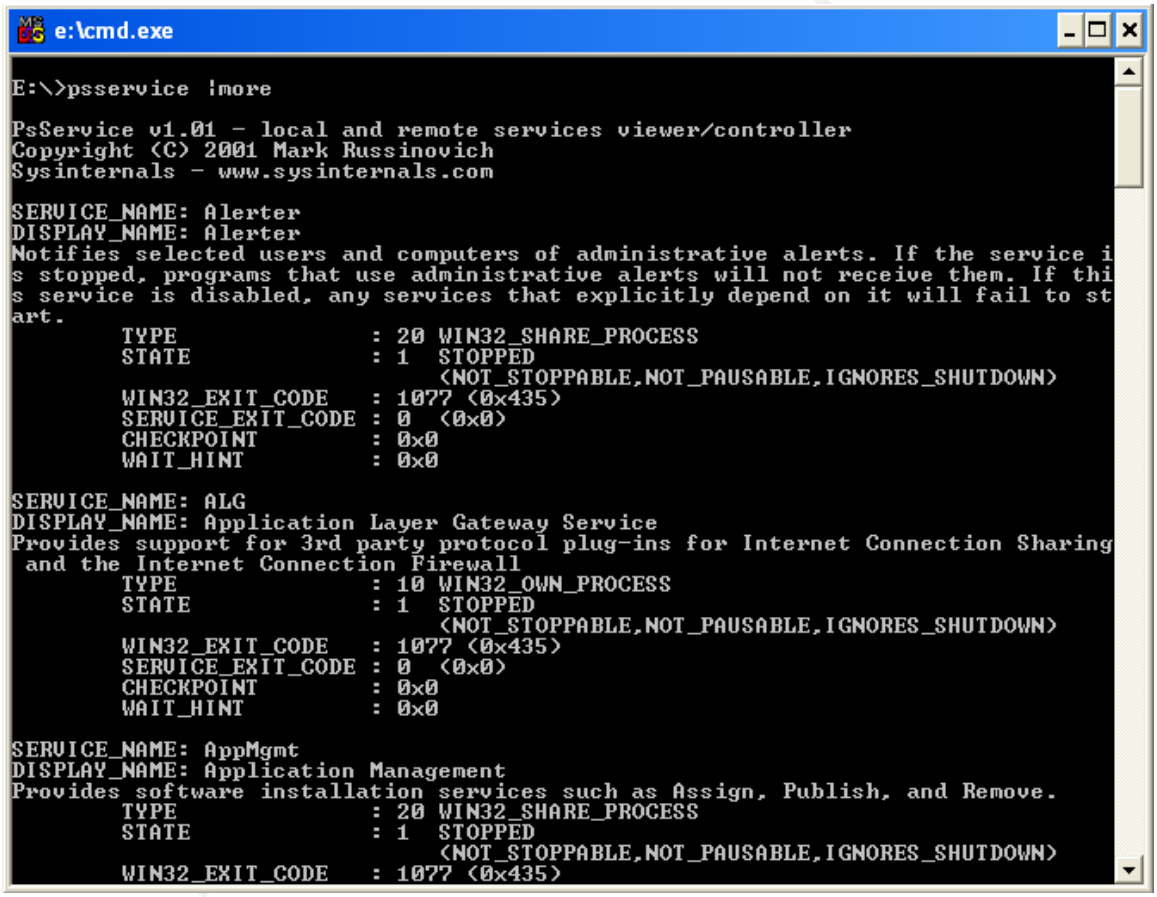


Figure A-7: psservice

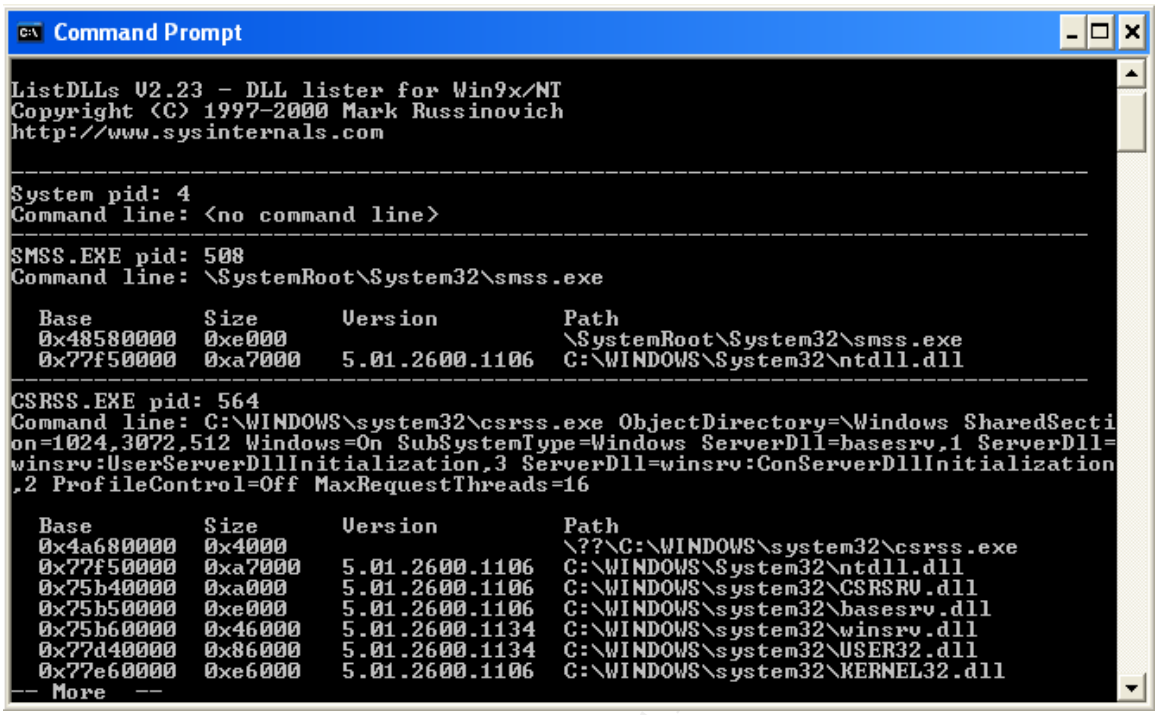


Figure A-8: listdlls

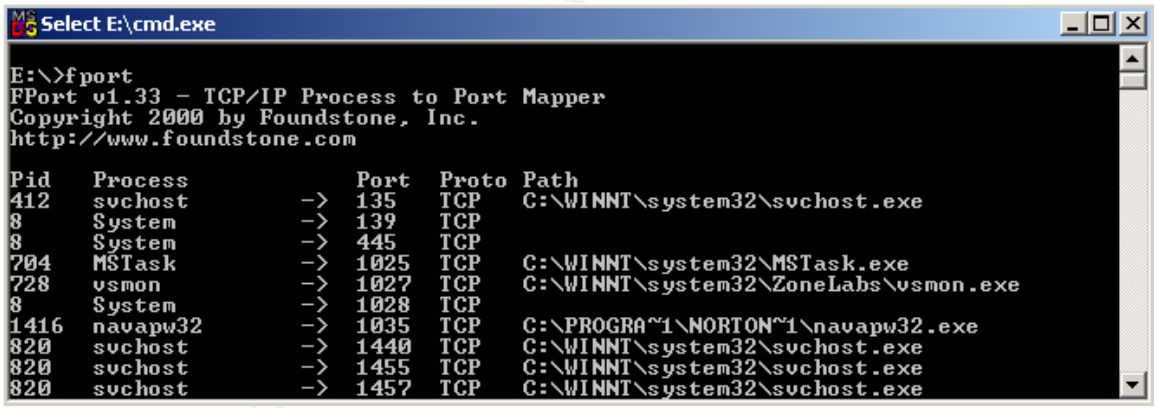


Figure A-9: fport

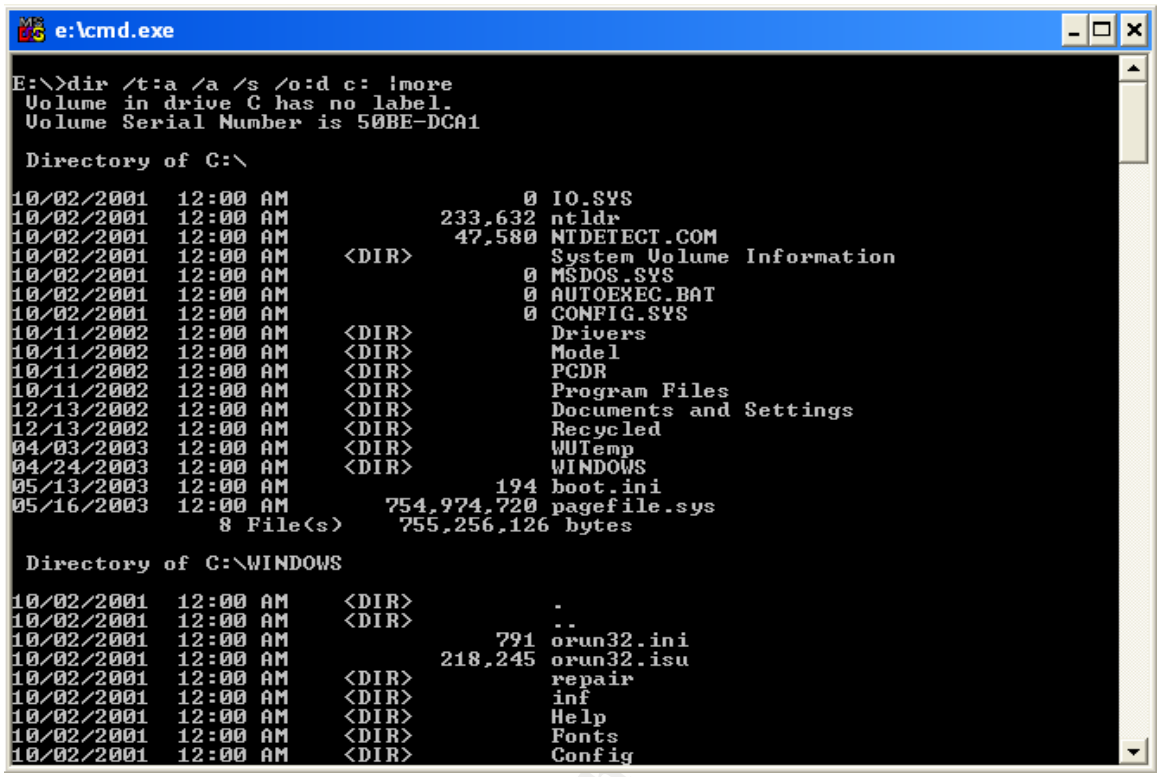


Figure A-10: Last Access Time

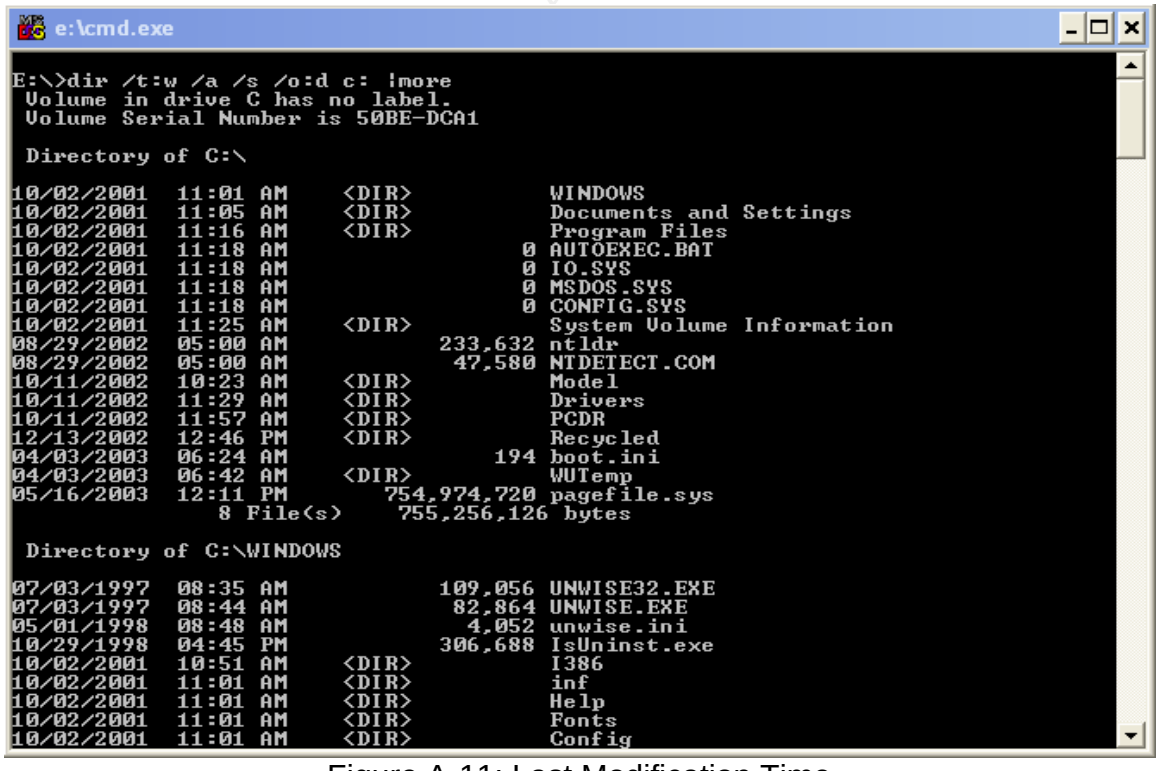


Figure A-11: Last Modification Time

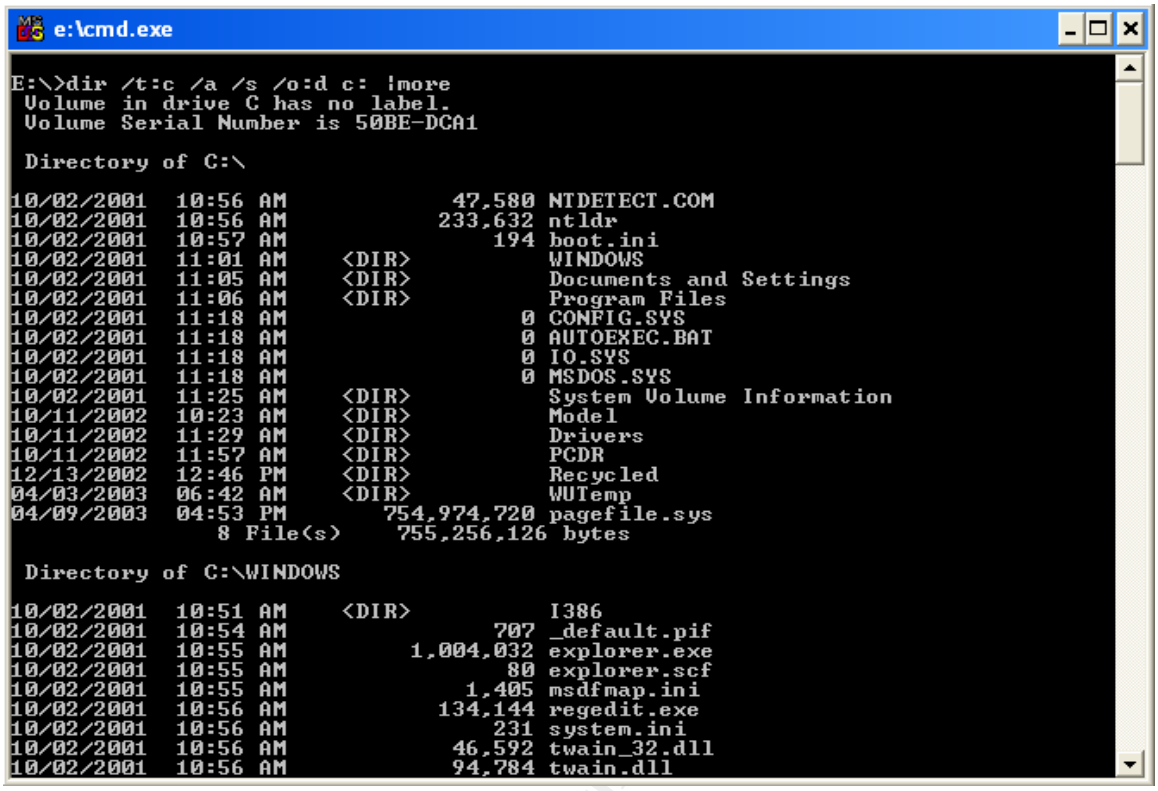


Figure A-12: Last Create Time

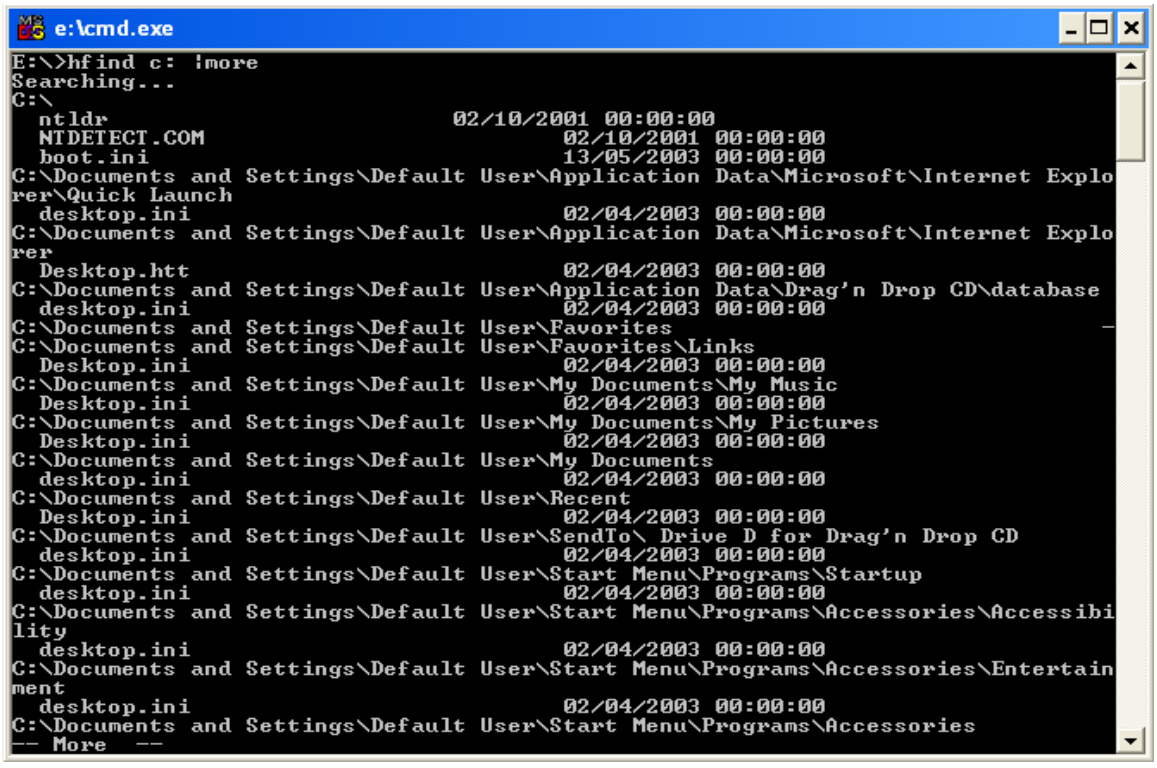


Figure A-13: hfind