Analyzing a Binary File and File Partitions for Forensic Evidence

Practical Assignment Version 1.2 (December 30, 2002)

By
James Michael Butler

Submitted 5/24/03 to SANS
in Partial Fulfillment of the Requirements
For the GIAC Certified Forensic Analyst (GCFA) Certification

Abstract/Summary

This paper Includes the following three parts: Part 1: Analyze an Unknown Binary; Part 2 – Option 1: Perform Forensic Analysis on a system; and Part 3: Legal Issues of Incident Handling.

The Binary being analyzed in Part 1 was provided on line at GIAC and represented an unknown program found on a compromised system.

The Author located a system, no longer being used, at his place of employment for Part 2. In that exercise, we will determine if there is any malware or any sign(s) of malicious acts on the drive in question.

Part 3 will be answers to various questions concerning US Federal law and Florida statutes as they pertain to computer Incident Handling.

Part 1: Analyze an Unknown Binary

Binary Details

- Name of file: target2.exe. This is a default name given by C compiler to compiled code – there is no particular significance to this name. Searches for this name on the Web resulted in nothing pertaining to this particular program. Most of the references were, in fact, concerning the use of a C compiler.
- The first line of text in the bintext listing below indicates that the program will not run in DOS. This identifies the program as a windows program. The .exe filename extension and the references to several DLL (Dynamic Linked Library) files are further proof that the exe will run under Windows. The references to services and the registry indicate Windows NT or above would be the target system. If this were a UNIX program, it most likely would not have the .exe extension and it certainly would not have any of the other references, unless they were being used to misdirect a forensics search. Running the exe will be final proof of these points.
- Name of program: (See results of bintext scan below.) The program identifies itself as “ICMP Backdoor v0.1,” but subsequent searches on the web with that phrase revealed no results.
- File/MAC Time information: Target2.exe was Accessed 5/10/03 (when I unzipped the file)
- Target2.exe was last Modified and Created on the same date and time: 2/20/03 at 12:45:48. That is probably the last date and time a compile was run on the source code.
- File owner(s) as revealed by filestat.exe (provided by SANS to the forensics class) are as follows:

```
Creation Time - 20/02/2003  12:45:48
Last Mod Time - 20/02/2003  12:45:48
Last Access Time - 20/05/2003  21:15:21
Main File Size - 26793
File Attrib Mask - Arch
Dump complete...Dumping \giac\target2.exe...
SD is valid.
SD is 92 bytes long.
SD revision is 1 == SECURITY_DESCRIPTOR_REVISION1
SD's Owner is Not NULL
SD's Owner-Defaulted flag is FALSE
  SID = BUILTIN/Administrators   S-1-5-32-544
SD's Group-Defaulted flag is FALSE
  SID = MBUTLER/None   S-1-5-21-572021508-1092557432-1361071929-513
SD's DACL is Present
SD's DACL-Defaulted flag is FALSE
    ACL has 1 ACE(s), 28 bytes used, 0 bytes free
    ACL revision is 2 == ACL_REVISION2
  SID = /Everyone   S-1-1-0
    ACE 0 is an ACCESS_ALLOWED_ACE_TYPE
    ACE 0 size = 20
    ACE 0 flags = 0x00
    ACE 0 mask = 0x001f01ff -R -W -X -D -DEL_CHILD -CHANGE_PERMS -TAKE_OWN
SD's SACL is Not Present
Stream 1:
  Type: Security
  Stream name = a Size: 92

Stream 2:
  Type: Data
  Stream name = a Size: 26793

Stream 3:
  Type: Unknown
  Stream name = a Size: 64
```

Based on the above information, it appears the file is owned by "Administrator," and the group SID is assigned to this author. If the file was zipped on an NTFS system before I downloaded it for analysis, it could be that the Administrator designation was valid. My group assignment was surely added when the file was unzipped onto my NTFS drive. To test this theory, I copied the file to a diskette in the FAT format, as opposed to NTFS. A filestat on that copy revealed the following:

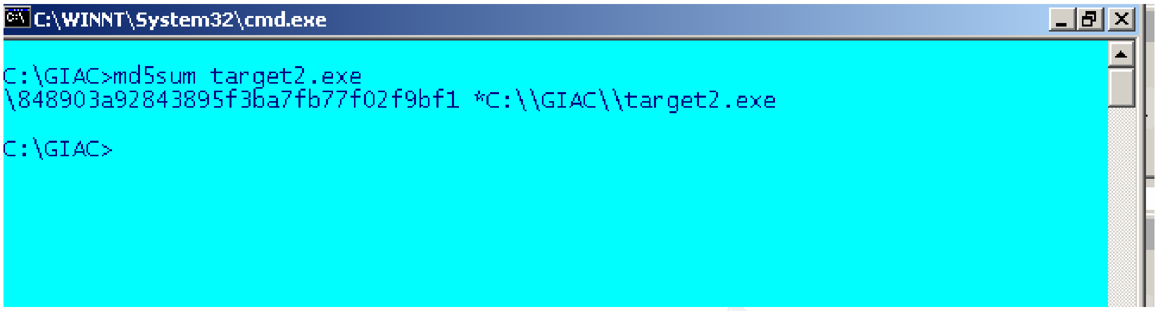
```
Creation Time - 22/05/2003  18:23:10
Last Mod Time - 20/02/2003  12:45:48
Last Access Time - 22/05/2003  00:00:00
Main File Size - 26793
File Attrib Mask - Arch
Dump complete...Dumping a:target2.exe...
SD is valid.
SD is 44 bytes long.
SD revision is 1 == SECURITY_DESCRIPTOR_REVISION1
SD's Owner is Not NULL
SD's Owner-Defaulted flag is FALSE
  SID = /Everyone   S-1-1-0
SD's Group-Defaulted flag is FALSE
  SID = /Everyone   S-1-1-0
SD's DACL is Present
SD's DACL-Defaulted flag is FALSE
SD has a NULL DACL explicitly specified      (allows all access to Everyone)
    This does not apply to this SD, but for comparison,
    a non-NULL DACL pointer to a 0-length ACL allows  no access to   anyone
SD's SACL is Present
SD's SACL-Defaulted flag is FALSE
SD has a NULL SACL explicitly specified
Stream 1:
  Type: Security
  Stream name = a Size: 44

Stream 2:
  Type: Data
  Stream name = a Size: 26793
```

In this case, the Owner and the Group are both listed as /Everyone, as expected. The significance of the ownership of this file is in question, while the ownership of a file actually discovered on a compromised system will be more important.

- The author of the code is identified as "spoof." This did not seem to match the listed authors of code that was found on the internet. When a search was done on the phrase appearing in the code: "Code by spoof" – results were negative.
- File size is 26,793 bytes. (Size on disk is 27,136 bytes. Size on disk varies from the actual file size due to the way a hard disk uses space. Information is stored on the hard drive in blocks of space that vary from disk to disk. The block size of my drive caused there to be slack space – (or unused space) – located at the end of the file in order to fill up the block. In this case, the difference between the actual file size and the size on disk will provide the size of the unused slack space: 343 bytes.)

- The MD5 hash of the file was obtained by the following command:
  md5sum target2.exe
- MD5 hash: \848903a92843895f3ba7fb77f02f9bf1 *C:\\GIAC\\target2.exe.
- We were not provided with the original md5 hash in the zip file with the target2.exe program for comparison.
- Screenshot of MD5:



```
C:\GIAC>md5sum target2.exe
\848903a92843895f3ba7fb77f02f9bf1 *C:\\GIAC\\target2.exe

C:\GIAC>
```

Key words associated with file

The following words and phrases were found in the target2.exe binary using bintext. Bintext.exe version 3.00 was installed from the SANS CD provided to the Forensics class. Bintext is a Windows program providing a graphical user interface for examining the content of files. After running BinText, I browsed to the file to be examined - target2.exe. Once selected, then I clicked on the "Go" button to scan the file. Once the strings are revealed, they can be saved by clicking on the "Save" button. In our case, we save the text to a file called "target2_bintext.txt" Here is the content of that file:

```
File pos    Mem pos      ID   Text
========    =======      ==   ====
0000004D    0040004D      0   !This program cannot be run in DOS mode.
000001D0    004001D0      0   .text
000001F8    004001F8      0   .rdata
0000021F    0040021F      0   @.data
00000248    00400248      0   .rsrc
000011D0    004011D0      0   D$,QPR
000011FC    004011FC      0   D$ j'P
0000121E    0040121E      0   T$,j'RP
000012FE    004012FE      0   T$,VRS
00001327    00401327      0   D$ j'P
00001349    00401349      0   T$,j'RP
00001408    00401408      0   L$ j'Q
0000142B    0040142B      0   D$,j'PQ
00001540    00401540      0   D$0QPR
0000156E    0040156E      0   D$$j'P
00001590    00401590      0   T$0j'RP
00001678    00401678      0   T$0URV
000016A1    004016A1      0   D$$j'P
000016C3    004016C3      0   T$0j'RP
00001803    00401803      0   D$ j'PQ
000019AF    004019AF      0   T$$QRj
000019CE    004019CE      0   D$$PW
00001BD6    00401BD6      0    h0A@
00001CEA    00401CEA      0   SPhxD@
00001D10    00401D10      0   SQhpD@
00001D65    00401D65      0   D$@SPS
00001E16    00401E16      0   T$|RP
00001E77    00401E77      0   USSSP3
00001F25    00401F25      0   D$(PQ
00002050    00402050      0   x!xu\
00002056    00402056      0   x"iuV
0000205C    0040205C      0   x#tuP
```

```
0000207A   0040207A      0   IQh@A@
00002270   00402270      0   t1h@D@
000022B4   004022B4      0   Ht Ht
0000243E   0040243E      0   Ph<B@
00002460   00402460      0   T$(QR
0000249D   0040249D      0   L$0PQ
00002528   00402528      0   Ph0C@
000032EA   004032EA      0   Sleep
000032F2   004032F2      0   HeapAlloc
000032FE   004032FE      0   GetProcessHeap
00003310   00403310      0   TerminateProcess
00003324   00403324      0   ReadFile
00003330   00403330      0   PeekNamedPipe
00003340   00403340      0   CloseHandle
0000334E   0040334E      0   CreateProcessA
00003360   00403360      0   CreatePipe
0000336E   0040336E      0   WriteFile
0000337A   0040337A      0   GetLastError
0000338A   0040338A      0   LocalAlloc
00003396   00403396      0   KERNEL32.dll
----------------------------------------------------------
000033A6   004033A6      0   StartServiceCtrlDispatcherA
000033C4   004033C4      0   SetServiceStatus
000033D8   004033D8      0   RegisterServiceCtrlHandlerA
000033F6   004033F6      0   CloseServiceHandle
0000340C   0040340C      0   ControlService
0000341E   0040341E      0   QueryServiceStatus
00003434   00403434      0   OpenServiceA
00003444   00403444      0   CreateServiceA
00003456   00403456      0   OpenSCManagerA
00003468   00403468      0   DeleteService
00003478   00403478      0   StartServiceA
00003488   00403488      0   ChangeServiceConfigA
000034A0   004034A0      0   QueryServiceConfigA
----------------------------------------------------------
000034B4   004034B4      0   ADVAPI32.dll
000034C4   004034C4      0   WSAIoctl
000034D0   004034D0      0   WSASocketA
000034DC   004034DC      0   WS2_32.dll
000034E8   004034E8      0   MFC42.DLL
000034F4   004034F4      0   memmove
00003506   00403506      0   fprintf
00003518   00403518      0   sprintf
00003522   00403522      0   perror
0000352C   0040352C      0   strstr
0000353E   0040353E      0   printf
00003546   00403546      0   MSVCRT.dll
00003554   00403554      0   __dllonexit
00003562   00403562      0   _onexit
0000356C   0040356C      0   _exit
00003574   00403574      0   _XcptFilter
00003582   00403582      0   __p___initenv
00003592   00403592      0   __getmainargs
000035A2   004035A2      0   _initterm
000035AE   004035AE      0   __setusermatherr
000035C2   004035C2      0   _adjust_fdiv
000035D2   004035D2      0   __p__commode
000035E2   004035E2      0   __p__fmode
000035F0   004035F0      0   __set_app_type
00003602   00403602      0   _except_handler3
00003616   00403616      0   _controlfp
00003624   00403624      0   ??0Init@ios_base@std@@QAE@XZ
00003644   00403644      0   ??1Init@ios_base@std@@QAE@XZ
00003664   00403664      0   ??0_Winit@std@@QAE@XZ
0000367C   0040367C      0   ??1_Winit@std@@QAE@XZ
00003692   00403692      0   MSVCP60.dll
00004049   00404049      0   ERROR 3
00004055   00404055      0   ERROR 2
00004061   00404061      0   ERROR 1
*0000406C   0040406C      0   impossibile creare raw ICMP socket
00004098   00404098      0   RAW ICMP SendTo:
```

```
*000040AE   004040AE      0   ======================== Icmp BackDoor V0.1
========================
*000040F4   004040F4      0   ========= Code by Spoof. Enjoy Yourself!
0000411E   0040411E      0    Your PassWord:
00004138   00404138      0   cmd.exe
00004142   00404142      0    Exit OK!
00004150   00404150      0   Local Partners Access
----------------------------------------------------------
0000416A   0040416A      0   Error UnInstalling Service
0000418A   0040418A      0   Service UnInstalled Sucessfully
000041B2   004041B2      0   Error Installing Service
000041CE   004041CE      0   Service Installed Sucessfully
000041F5   004041F5      0   Create Service %s ok!
0000420D   0040420D      0   CreateService failed:%d
00004229   00404229      0   Service Stopped
0000423D   0040423D      0   Force Service Stopped Failed%d
00004260   00404260      0   The service is running or starting!
00004288   00404288      0   Query service status failed!
000042A8   004042A8      0   Open service failed!
000042C1   004042C1      0   Service %s Already exists
000042DC   004042DC      0   Local Printer Manager Service
000042FC   004042FC      0   smsses.exe
00004309   00404309      0   Open Service Control Manage failed:%d
00004338   00404338      0   Start service successfully!
00004358   00404358      0   Starting the service failed!
00004378   00404378      0   starting the service <%s>...
00004398   00404398      0   Successfully!
000043A8   004043A8      0   Failed!
000043B4   004043B4      0   Try to change the service's start type...
000043E0   004043E0      0   The service is disabled!
000043FC   004043FC      0   Query service config failed!
----------------------------------------------------------
000062DB   004062DB      0   ?????
*00005064   00405064      0   Hello from MFC!
000060F3   004060F3      0   \winnt\system32\smsses.exe
00006181   00406181      0   \winnt\system32\smsses.exe
000062B3   004062B3      0   \\199.107.97.191\C$
0000632F   0040632F      0   \winnt\system32
000063A7   004063A7      0   \winnt\system32\reg.exe
0000642F   0040642F      0   \winnt\system32\reg.exe
000064B7   004064B7      0   \winnt\system32\reg.exe
0000653F   0040653F      0   \winnt\system32\reg.exe
000065BD   004065BD      0   \winnt\system32\reg.exe
00006645   00406645      0   \winnt\system32\reg.exe
000066CD   004066CD      0   \winnt\system32\reg.exe
00006755   00406755      0   \winnt\system32\reg.exe
000067DD   004067DD      0   \winnt\system32\reg.exe
00005062   00405062      1   Hello from MFC!
```

Program Description

Web searches were initiated on the phrases marked above in the bintext results with an *. None of those searches revealed source code for this program. There was some code located that may have provided snippets to the author of our code. This was located searching for the foreign phrase: "impossibile creare raw" which appeared in the bintext results. This caused one site to be listed under Google: http://www.s0ftpj.org/bfi/online/bfi7/bfi07-13.html. This site did host code for what appears to be an ICMP tunneling program. However, the author was different than "spoof," and the code contained information I would have expected to have seen above, and did not contain some of the information revealed by bintext. For example, the phrase/line:

```
*000040F4   004040F4      0   ========= Code by Spoof. Enjoy Yourself!
```

does not appear in the code listed at the above site. Therefore, the source code was not located on the internet.

The bintext results above also indicated a number of statements related to services. In fact, judging only from that data, one might assume one of this binary's purposes in life is remotely starting and stopping services on a compromised computer. File positions 33a6 to 34a0 list 13 different action statements referring to starting, stopping, querying, or otherwise manipulating services. Look for dashed lines in the text information indicating start and stop of services information. Twenty-four more message statements are listed from file positions 416A to 43FC, also marked with dashed lines. In addition, starting at file position 63A7, there are nine references to reg.exe. These imply the use of reg.exe by this program to make changes in the registry. Reg.exe is not included with the Windows install. Reg.exe is used, with appropriate parameters, to query or make changes in the Windows Registry on the fly from a DOS prompt. This makes it simple to script such changes to happen automatically. Because the registry controls so many functions of the computer, we will have to observe carefully to see if registry changes are made by this software. It must be noted that registry changes could also be made without the use of Reg.exe. I am puzzled, actually, as to why the author of the code would even resort to or refer to Reg.exe.

Target2.exe appears to be, based on the strings above and as mentioned above, some sort of code designed to remotely control services and/or the registry on a computer running Windows NT, 2000, and/or XP. I am basing that opinion on the number of string references in the code to controlling services. However, as we executed the program in an isolated environment as noted elsewhere in this document, no communication took place that might have had that effect. Based on the sketchy information revealed by bintext, I would have to execute the code to verify its purpose. I arranged to use a "loaner" from LAN/Desktop with the understanding that they would use Ghost to copy the hard drive so that it could be put back like it was before I proceeded to trash it. See the process followed to run and examine the code elsewhere in this document.

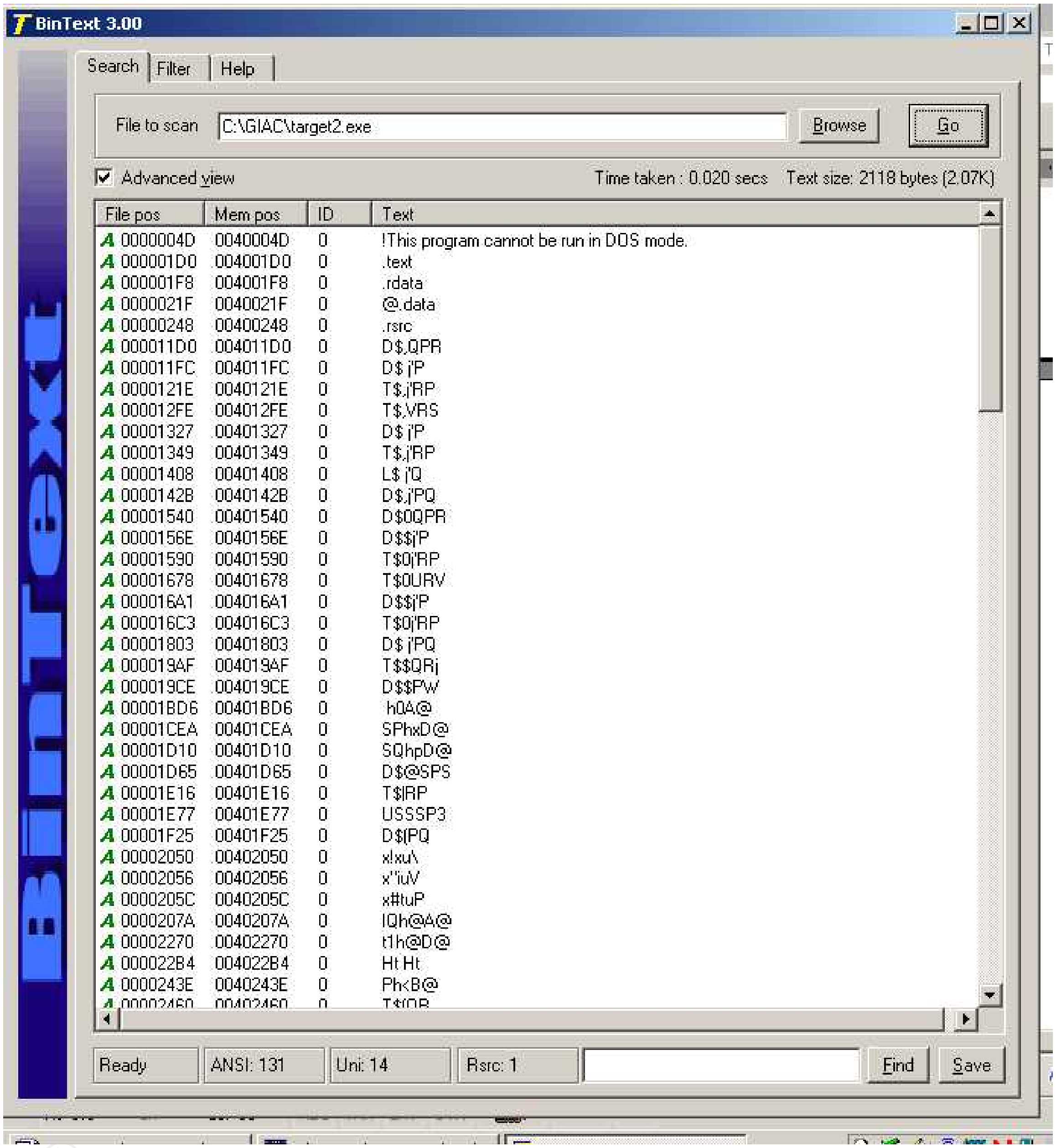
Target2.exe also appears by its language to utilize the ICMP protocol to create a back door for that purpose. Even though the reference is made to "ICMP Backdoor v0.1," it is not clear that the program would really provide access over ICMP. It could be that the title as well as the single reference to Loki, a true ICMP back door program, may have been used to throw off forensics investigators.

The last time this particular binary (target2.exe) was used would have to be on or before the last accessed date and time. As noted above, Target2.exe was Accessed 5/10/03 (when I unzipped the file). Target2.exe was last Modified and Created on the same date and time: 2/20/03 at 12:45:48. That is probably the last date and time a compile was run on the source code.

The steps I followed to determine the above information:

1. I ran strings and bintext on target2 program. For Strings, I used the command: strings target2.exe
   For bintext, I simply typed the command bintext as a DOS prompt to get the graphical user interface (GUI). Then I clicked on Browse to locate my

file, then clicked on Go to do the scan. To save the data in a text file, I clicked on File/Save As. That file was imported into this document above. Here is a screen shot of BinText 3.0:



File to scan C:\GIAC\target2.exe

☑ Advanced view   Time taken : 0.020 secs   Text size: 2118 bytes (2.07K)

```
  File pos   Mem pos   ID   Text
A 0000004D  0040004D   0    !This program cannot be run in DOS mode.
A 000001D0  004001D0   0    .text
A 000001F8  004001F8   0    .rdata
A 0000021F  0040021F   0    @.data
A 00000248  00400248   0    .rsrc
A 000011D0  004011D0   0    D$,QPR
A 000011FC  004011FC   0    D$ j'P
A 0000121E  0040121E   0    T$,j'RP
A 000012FE  004012FE   0    T$,VRS
A 00001327  00401327   0    D$ j'P
A 00001349  00401349   0    T$,j'RP
A 00001408  00401408   0    L$ j'Q
A 0000142B  0040142B   0    D$,j'PQ
A 00001540  00401540   0    D$0QPR
A 0000156E  0040156E   0    D$$j'P
A 00001590  00401590   0    T$0j'RP
A 00001678  00401678   0    T$0URV
A 000016A1  004016A1   0    D$$j'P
A 000016C3  004016C3   0    T$0j'RP
A 00001803  00401803   0    D$ j'PQ
A 000019AF  004019AF   0    T$$QRj
A 000019CE  004019CE   0    D$$PW
A 00001BD6  00401BD6   0     h0A@
A 00001CEA  00401CEA   0    SPhxD@
A 00001D10  00401D10   0    SQhpD@
A 00001D65  00401D65   0    D$@SPS
A 00001E16  00401E16   0    T$|RP
A 00001E77  00401E77   0    USSSP3
A 00001F25  00401F25   0    D$|PQ
A 00002050  00402050   0    x!xu\
A 00002056  00402056   0    x"iuV
A 0000205C  0040205C   0    x#tuP
A 0000207A  0040207A   0    |Qh@A@
A 00002270  00402270   0    t1h@D@
A 000022B4  004022B4   0    Ht Ht
A 0000243E  0040243E   0    Ph<B@
A 00002460  00402460   0    T$(QR
```

Ready   ANSI: 131   Uni: 14   Rsrc: 1

Since bintext offered more information – such as an IP address (explained later) – I redirected the output of bintext to a file called bintext.txt. The content of that file is displayed above as key words. The strings included in the program give many clues as to its purpose, although one must be careful in interpretation as hackers can include text designed to throw off the forensics specialist. By including the name(s) of programs that do something altogether different than the suspect file, a specialist may initially determine that a file may be a particular program or other. For example, the strings in this program included the name “loki.” While there is a program named loki that is a back door using the ICMP (ping) protocol, it is doubtful that this program is the same one. In this case, this author suspects the word “loki” was inserted to throw off the investigator. Actually, in context, loki could be construed to be a possible password.

2. Yahoo searches were performed using a number of strings found in the code to try to determine the location of the source code. They were largely unsuccessful, as noted above.
3. The previously mentioned LAN/Desktop PC acquired for this test was an IBM desktop. The specifications are as follows:
   IBM Personal Computer 300 PL M/T 6565 E2U with a PIII 300 MHZ processor
   Fujitsu MPE3204AH B6 hard drive (approx. 20 gb) partitioned into two drives – C and D – each with 9.49 gb of total space
   One CD: LG CD-ROM CRD-8480C DVD/CD-R
   640 MB of Random Access Memory
   One IBM 10/100 Ether Jet PCI Adapter with Alert on LAN
   USB Ports
   S3 Inc., Savage 4 Video Adapter
   MS Windows 2000 Professional version 5.00.2195 Service Pack 2
   This computer was ideal, in one way, for this experiment since it had both a C: and D: drive. I was able to put forensics tools on the D: drive and also to write results from the experiments on the D: drive so as not to disturb the C: drive more than is necessary.
4. I installed a number of monitoring programs on the loaner D: drive so that we could watch what happens to the registry (regmon.exe), the files (filemon.exe), the ports (TDIMon.exe), and the security access requests (TokenMon.exe). Programs I installed include but are not limited to the following.
   a. DumpACL (to record the current status of access control on the computer) Ran prior to running target2.exe and dumped data to a file called DumpACL.txt.
   b. DumpREG (to record the current status of the Registry)
   c. Ethereal (to sniff traffic to and from the compromised computer)
   d. NTFileMon (monitor use of and/or changes in files and their permissions)
   e. REGMON (to monitor changes/updates to the registry)
   f. PortMon (to monitor the use/attempted use of serial communication or parallel printer ports)
   g. Procexp (Process explorer with a graphical interface to show what processes are tied to what files – will note if new processes are started by target2.exe.)
   h. Pslist.exe (Command line utility that can be set up to dump the process list into a text file for future reference on a timed basis – default of 1 second intervals - in “Task Manager” mode.)

   i. TDIMon (To monitor usage of TCP/UDP ports – particularly by the rogue software.)
   j. TokenMon (to Monitor security events in real time showing what program is attempting to achieve what level of security.)
   k. TDS3 (To scan for the existence of Trojans on the compromised box after we run our software.)

5. In addition to the loaner, I also set up my Linux forensics computer to run Ethereal so scanning could be done from the network. (The Linux forensics computer is described later in this document in conjunction with the forensics study of a “compromised” system.)
6. A crossover cable was used to connect the Windows 2k (compromised) box to the Linux forensics computer. This has the effect of isolating the compromised computer from the network, but still allows us to monitor network activity of the Windows computer with the Linux computer.
7. I wrote the following script with a text editor in order to kick off a series of events prior to the forensics study of the target2.exe program:

```
@echo off
cls
rem SaveMachineStatus.bat Script written for GIAC exercise by J. Michael Butler
rem Script expects tools to be located in d:\giac\giac\tools directory
rem They could be anywhere as long as the script knows where to find them
rem Could add another parameter (%2) to identify location of tools for different computers
echo This script will run several programs to prepare for Forensics Testing of suspect binary
echo on this Windows computer...  Several snapshots will be taken of the status of the computer
echo and the pslist process will be started to capture process activity while the program is running.
echo.
echo.
rem First I need to see if some parm was entered indicating where the results would be stored
echo Checking for parms...
if %1 == "" then goto error

rem %1 = the output directory for all the results of these programs
echo You have chosen to save all results to the %1 directory.  If this is not correct,
echo Use CTRL+C to abort now or
pause

rem create the directories if they don’t exist yet
if exist %1 goto skipcreatedir
echo Creating Output Directories…
md %1
md %1\winnt
md %1\winnt\system
md %1\winnt\system32
md %1\winnt\system32\drivers
md %1\winnt\system32\drivers\etc
:skipcreatedir

rem Backup system files for later comparison.
echo Backing up system files - winnt, system, system32, drivers, etc...
xcopy %systemroot%\*.* %1\winnt\ /h
xcopy %systemroot%\system\*.* %1\winnt\system\ /h
xcopy %systemroot%\system32\*.* %1\winnt\system32\ /h
xcopy %systemroot%\system32\drivers\*.* %1\winnt\system32\drivers\ /h
xcopy %systemroot%\system32\drivers\etc\*.* %1\winnt\system32\drivers\etc\ /h

echo Creating DIR of all DLL files on C: drive...
rem Command starts in the root directory and /s tells it to look in all subdirectories.  /o=n
rem will cause the directories to be sorted in order by name.  > %1\dlldir.txt will cause
rem the results to be “piped” to the file dlldir.txt in the directory specified by the user.
dir c:\*.dll /s/o=n > %1\dlldir.txt

rem %systemroot% is an environment variable that specifies the name of the system directory.
rem When the user runs the script, all occurrences of %systemroot% will be replaced
rem with WINNT, for example, on a typical Windows 2000 or NT computer.
echo creating DIR of all %systemroot% files...
dir %systemroot% > %1\systemdir.txt

rem Every file in several directories will be hashed into a value that can be compared later to make sure
rem none of the files has changed.  In each command, the results will be piped into the specified file.
rem It must be noted that “busy” files will not be accessible for the md5 sum.
echo Creating md5sums of all system files in system root, system, and system32...
d:\giac\giac\tools\md5sum \*.* > %1\rootdirmd5.txt
d:\giac\giac\tools\md5sum %systemroot%\*.* > %1\systemrootdirmd5.txt
d:\giac\giac\tools\md5sum %systemroot%\system\*.* > %1\systemdirmd5.txt
d:\giac\giac\tools\md5sum %systemroot%\system32\*.* > %1\system32dirmd5.txt
d:\giac\giac\tools\md5sum %systemroot%\system32\drivers\*.* > %1\driversdirmd5.txt
d:\giac\giac\tools\md5sum %systemroot%\system32\drivers\etc\*.* > %1\etcdirmd5.txt

rem A series of DumpACL commands are issued to get text copies of the ACL (Access Control List)
rem policies, services, shares, users, registry entries for the Local Machine hive, and for the Users
rem hive, the Groups, and the Rights settings for the computer.
echo DumpACL Policy...
d:\giac\giac\tools\dumpacl /rpt=policy /outfile=%1\DumpAclPolicyTab.txt /saveas=tsv
echo DumpACL Services...
d:\giac\giac\tools\dumpacl /rpt=services /outfile=%1\DumpAclServicesTab.txt /saveas=tsv
echo DumpACL Shares...
d:\giac\giac\tools\dumpacl /rpt=shares /outfile=%1\DumpAclSharesTab.txt /saveas=tsv
echo DumpACL Users...
d:\giac\giac\tools\dumpacl /rpt=users /outfile=%1\DumpAclUsersTab.txt /saveas=tsv
echo DumpACL Registry - Local Machine...
d:\giac\giac\tools\dumpacl /rpt=registry=hkey_local_machine /outfile=%1\DumpAclRegistryLTab.txt /saveas=tsv
echo DumpACL Registry - Users...
d:\giac\giac\tools\dumpacl /rpt=registry=hkey_users /outfile=%1\DumpAclRegistryUTab.txt /saveas=tsv
echo DumpACL Groups...
d:\giac\giac\tools\dumpacl /rpt=groups /outfile=%1\DumpAclGroupsTab.txt /saveas=tsv
echo DumpACL Rights...
d:\giac\giac\tools\dumpacl /rpt=rights /outfile=%1\DumpAclRightsTab.txt /saveas=tsv

echo End of run.
echo.
echo.
rem I inserted a reminder here of all the programs I need to run so I don’t overlook something.
echo Now starting pslist to run during test:  (press esc to abort pslist)
echo Don’t forget to run nmap on the Forensics computer before and after compromise.
echo You will need to run filemon and regmon and apply filter in regmonfilter.txt file.
echo Also run TDIMon and TokenMon.
Pause

rem the following program will take snapshots of the processes every 1 second and dump the results
rem into the specified file, “pslist.txt.”
d:\giac\giac\tools\pslist -s > %1\pslist.txt

goto end

:error
rem This will only pop up if there are no parameters entered with the command.
echo You must specify the directory where you want the results stored.  Use different directory every
echo time in order to not overwrite previous results.
echo.
echo.
echo Correct syntax:  SaveMachineStatus dirname
echo.
echo.
goto end

:end
rem I added a pause so the window would not disappear as soon as the script was finished.
rem That way the user can see any error messages that might still be on the screen.
Pause
```

8. The script will make a copy of all files in C:\WINNT to the WINNT subdirectory it creates off of the directory specified in the command line.
9. In the same way copies of C:\WINNT\SYSTEM, C:\WINNT\SYSTEM32, C:\WINNT\SYSTEM32\DRIVERS, and the C:\WINNT\SYSTEM32\DRIVERS\ETC directories are placed in subdirectories off of that WINNT directory named SYSTEM, SYSTEM32, SYSTEM32\DRIVERS, and SYSTEM32\DRIVERS\ETC, respectively. These files were all copied so a file compare using fc.exe could be done to see if any of them changed during the running of target2.exe.
10. By using a script to create snapshots, I could do so in a timely manner so any changes to status might be attributed to the target2.exe program. The snapshots taken can be seen by reviewing the comments (rem statements) and the echoed (echo text to be echoed) statements in the above script.

After the creation of the above script, which I named SaveMachineStatus.bat, I created two shortcuts to the script. Altering the properties of the shortcuts, I added the parameter: d:\before
to one of the command lines, and d:\after
to the command line of the second shortcut. Then renaming the shortcuts to include “Before” or “After” as appropriate, I was ready to run the script twice. The first time I run it with the “Before” parameter, it will create a subdirectory called d:\before. Then it will dump all the results of testing into that directory. After the target2.exe program has run and I am through testing, I will run the “After” script to create a directory with the current status of the test computer.

The changes made to the short cuts were as follows: Right click on the short cut, then click on properties. Going to the end of the “target” line, add a space and the parameter specifying the destination directory. Then click on Apply and OK to save the changes. Renaming the shortcuts is simply a matter of right clicking on the shortcut and choosing the “Rename” option.

Keep in mind that the script can be run from a command prompt as well. In that case the short cuts and their modification would not be necessary. The user would simply type the script name followed by the directory where the results are to be directed. (Ex. savemachinestatus d:\pathname where d=the desired drive and pathname=the path to the directory on that drive without the final backslash.)

Because the script would not execute the programs in multiple threads, I have built in a reminder of the other programs I wish to run before executing the target2.exe code. I created shortcuts to those programs and placed the shortcuts all in one directory so that it will be easy to start up the programs.

Here is the chronology followed for testing the application:

1. Preparation steps taken as listed above.
2. On the Linux computer, I ran the /sbin/ifconfig program to find the IP information needed to configure the compromised box.
3. I then manually set the IP address of the test computer one digit away from the Linux computer with the same subnet mask so they could communicate if connected. I did that by right clicking on My Network

Places, selecting Properties, selecting Local Area Connection, Clicking on Properties button, then selecting and configuring Internet Protocol (TCP/IP) by clicking on that properties button. The General tab allowed me to change the IP address, subnet mask to match the Linux computer, and to set the default gateway to the IP address of the Linux computer.

4. Connected a crossover cable from the “compromised” computer to the Linux forensics computer so they could communicate.
5. Started Ethereal on Linux computer, a sniffer set up to capture all network packets to/from the “compromised” computer. Ethereal was started by going to the directory where the Ethereal program resides, and typing the following command: ./ethereal
6. After Ethereal is started, the user is presented with an options window. In the Ethereal window, I left the default settings on, and also selected settings to update the packets real time in the Ethereal window. Once these settings are selected, the user just clicks on OK and network packets, assuming there are any, can be seen on the Linux screen.
7. Opened a command window with Start/Run/cmd and pinged Linux box and watched to make sure the ping was detected by watching the results in Ethereal.
8. Opened a command window with Start/Run/CMD and copied over the target2.exe to the c:\ directory from a floppy disk with the command: copy a:\target2.exe executed while in the c:\ directory.
9. Typed in the command target2 but did not press enter.
10. Opened a new explorer window with my six shortcuts in it for the programs I wish to run after my start up procedures.
11. Started my SaveMachineStatus BEFORE script, but did not touch a key “…to continue” after the first pause because I was not ready for the tests to begin.
12. Synchronized the clocks between the test computer and the Linux computer.
13. Went back and kicked off my script to save a snapshot of the ACL, Registry, and other system information as noted in the above script. The last thing that runs in the script is the Pslist dumping of process information in “Task Manager” mode into a text file in the d:\before subdirectory. That process will continue until the user hits ESC. However, as you can see from the above script, the pslist command is after a pause command, so the script will be paused waiting for “any” key to start the pslist command.
14. Before I “touched any key” to continue to the Pslist command, I ran nmap to get the open ports before compromise, and the six programs: Filemon, Portmon, Procexp, Regmon, Tdimon, and Tokenmon.

nmap was executed from a cmd (C:\>) prompt on the Linux server. The command executed was:

```
nmap –vv –sS –sU -10.xx.xx.xx > giacportsbefore.txt
```

which caused the “verbose” response of TCP and UDP port scans to be piped into the file named giacportsbefore.txt. As this box was not "hardened," a number of normal ports were open. Here are the results of the scan:

```
Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Host  (10.48.19.239) appears to be up ... good.
Initiating SYN Stealth Scan against  (10.48.19.239)
Adding TCP port 139 (state open).
Adding TCP port 135 (state open).
Adding TCP port 1026 (state open).
Adding TCP port 445 (state open).
The SYN Stealth Scan took 4 seconds to scan 1563 ports.
Initiating UDP Scan against  (10.48.19.239)
The UDP Scan took 9 seconds to scan 1563 ports.
Interesting ports on  (10.48.19.239):
(The 3114 ports scanned but not shown below are in state: closed)
Port       State       Service
135/tcp    open        loc-srv
135/udp    open        loc-srv
137/udp    open        netbios-ns
138/udp    open        netbios-dgm
139/tcp    open        netbios-ssn
401/udp    open        ups
402/udp    open        genie
445/tcp    open        microsoft-ds
445/udp    open        microsoft-ds
500/udp    open        isakmp
1025/udp   open        blackjack
1026/tcp   open        nterm


Nmap run completed -- 1 IP address (1 host up) scanned in 33 seconds
```

15. Executed the program target2.exe. After a pause of 15 seconds or so, with no screen activity or any other indication the program was running, I was returned to a C:\ > prompt.
16. I had seen no activity on Ethereal, (on the network), even though I expected some, so I went back to a DOS prompt and pinged the Linux box again. The connection was still good.
17. After waiting for a couple of minutes, I stopped the pslist process knowing that the data was recorded in the d:\before directory. (I actually copied the pslist data to a d:\during directory to facilitate comparisons of files in the before and after directories.)

Then I ran my "after" script. I closed down regmon after saving the data in order to facilitate the dump of the registry. The registry monitoring program showed the following hives were accessed by target2.exe, but they do not show any changes made to the registry. The hives accessed or attempted to access are included in the following excerpt from the regmon log:

```
9084    56.86640313    cmd.exe:276    OpenKey
        HKLM\System\CurrentControlSet\Control\Session Manager\AppCertDlls   NOTFOUND

9085    56.86646236    cmd.exe:276    OpenKey HKLM\Software\Microsoft\Windows
NT\CurrentVersion\Image File Execution Options\target2.exe  NOTFOUND
9086    56.86759071    target2.exe:264        OpenKey HKLM\Software\Microsoft\Windows
NT\CurrentVersion\Image File Execution Options\target2.exe  NOTFOUND
9087    56.86762955    target2.exe:264        OpenKey HKLM\Software\Microsoft\Windows
NT\CurrentVersion\Image File Execution Options\target2.exe  NOTFOUND
9088    56.86781057    target2.exe:264        OpenKey HKLM\Software\Microsoft\Windows
NT\CurrentVersion\Image File Execution Options\target2.exe  NOTFOUND
9089    56.87646139    target2.exe:264        OpenKey HKLM\Software\Microsoft\Windows
NT\CurrentVersion\Image File Execution Options\target2.exe  NOTFOUND
9090    56.87761294    target2.exe:264        OpenKey HKLM\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Winlogon    SUCCESS Key: 0xE2D7A480
```

```
9091    56.87765037    target2.exe:264        QueryValue
        HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\LeakTrack
        NOTFOUND
9092    56.87769786    target2.exe:264        CloseKey
        HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon  SUCCESS Key:
0xE2D7A480
9093    56.87775038    target2.exe:264        OpenKey HKLM    SUCCESS Key: 0xE2D7A480

9094    56.87778503    target2.exe:264        OpenKey HKLM\Software\Microsoft\Windows
NT\CurrentVersion\Diagnostics NOTFOUND
9095    56.87839907    target2.exe:264        OpenKey
        HKLM\System\CurrentControlSet\Control\Error Message Instrument\     NOTFOUND

9096    56.87876671    target2.exe:264        OpenKey HKLM\Software\Microsoft\Windows
NT\CurrentVersion\Compatibility32     SUCCESS Key: 0xE2D93120
9097    56.87880583    target2.exe:264        QueryValue
        HKLM\Software\Microsoft\Windows NT\CurrentVersion\Compatibility32\target2
        NOTFOUND
9098    56.87883739    target2.exe:264        CloseKey
        HKLM\Software\Microsoft\Windows NT\CurrentVersion\Compatibility32   SUCCESS Key:
0xE2D93120
9099    56.87890109    target2.exe:264        OpenKey HKLM\Software\Microsoft\Windows
NT\CurrentVersion\Compatibility2      SUCCESS Key: 0xE2D93120
9100    56.87896143    target2.exe:264        QueryValue
        HKLM\Software\Microsoft\Windows NT\CurrentVersion\Compatibility2\target20.0
        NOTFOUND
9101    56.87899021    target2.exe:264        CloseKey
        HKLM\Software\Microsoft\Windows NT\CurrentVersion\Compatibility2    SUCCESS Key:
0xE2D93120
9102    56.87904524    target2.exe:264        OpenKey HKLM\Software\Microsoft\Windows
NT\CurrentVersion\IME Compatibility   SUCCESS Key: 0xE2D93120
9103    56.87907430    target2.exe:264        QueryValue
        HKLM\Software\Microsoft\Windows NT\CurrentVersion\IME Compatibility\target2
        NOTFOUND
9104    56.87910363    target2.exe:264        CloseKey
        HKLM\Software\Microsoft\Windows NT\CurrentVersion\IME Compatibility SUCCESS Key:
0xE2D93120
9105    56.87939138    target2.exe:264        OpenKey
        HKLM\System\CurrentControlSet\Control\Session Manager\AppCompatibility\target2.exe
        NOTFOUND
9106    56.87944753    target2.exe:264        OpenKey HKLM\Software\Microsoft\Windows
NT\CurrentVersion\Windows      SUCCESS Key: 0xE2D93120
9107    56.87947463    target2.exe:264        QueryValue
        HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows\AppInit_DLLs     SUCCESS
        ""
9108    56.87951932    target2.exe:264        CloseKey
        HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows   SUCCESS Key:
0xE2D93120
9109    56.88031356    target2.exe:264        OpenKey HKCU    SUCCESS Key: 0xE2D93120
9110    56.88036357    target2.exe:264        OpenKey
        HKLM\System\CurrentControlSet\Control\Nls\MUILanguages      NOTFOUND
9111    56.88040882    target2.exe:264        OpenKey HKCU\Control Panel\Desktop    SUCCESS
        Key: 0xE2D94120
9112    56.88044905    target2.exe:264        QueryValue     HKCU\Control
Panel\Desktop\MultiUILanguageId       NOTFOUND
9113    56.88047587    target2.exe:264        CloseKey       HKCU\Control Panel\Desktop
        SUCCESS Key: 0xE2D94120
9114    56.88050129    target2.exe:264        CloseKey       HKCU    SUCCESS Key:
0xE2D93120
9115    56.88271722    target2.exe:264        OpenKey
        HKLM\System\CurrentControlSet\Control\ServiceCurrent SUCCESS Key: 0xE2D93120
9116    56.88274962    target2.exe:264        QueryValue
        HKLM\System\CurrentControlSet\Control\ServiceCurrent\(Default)      SUCCESS 0xB
9117    56.88280997    target2.exe:264        CloseKey
        HKLM\System\CurrentControlSet\Control\ServiceCurrent SUCCESS Key: 0xE2D93120

~pslist references removed between 9117 and 9978

9978    71.88405202    target2.exe:264        CloseKey       HKLM    SUCCESS Key:
0xE2D7A480
```

18. The reader will note that all of the above references show that target2.exe does not write to the registry. It seems to do a lot of reading and queries. This would lead one to believe it could be collecting information. However Filemon seems to indicate that the program never does anything with the information it collects – such as writing it to a disk. Here is an excerpt from the Filemon log showing where target2.exe was run from within a cmd prompt, then access to a list of DLL files was achieved by target2.exe. Note that there are no "Write" statements – only open, read, query type statements. Also note that the program is smart enough to look in several directories for the target DLL until it finds it:

```
2:43:44 AM     cmd.exe:704    IRP_MJ_CREATE  D:\Giac\GIAC\Binary\target2.exe       SUCCESS
       Attributes: Any Options: Open
2:43:44 AM     cmd.exe:704    IRP_MJ_CREATE  D:\Giac\GIAC\Binary\target2.exe       SUCCESS
       Attributes: N Options: Open
2:43:44 AM     cmd.exe:704    IRP_MJ_QUERY_INFORMATION
       D:\Giac\GIAC\Binary\target2.exe       SUCCESS FileInternalInformation
2:43:44 AM     cmd.exe:704    IRP_MJ_CLEANUP D:\Giac\GIAC\Binary\target2.exe       SUCCESS

2:43:44 AM     cmd.exe:704    IRP_MJ_CLOSE   D:\Giac\GIAC\Binary\target2.exe       SUCCESS

2:43:44 AM     cmd.exe:704    IRP_MJ_CREATE  D:\Giac\GIAC\Binary\target2.exe       SUCCESS
       Attributes: N Options: Open
2:43:44 AM     cmd.exe:704    FASTIO_QUERY_STANDARD_INFO
       D:\Giac\GIAC\Binary\target2.exe       SUCCESS Size: 26793
2:43:44 AM     cmd.exe:704    IRP_MJ_CLEANUP D:\Giac\GIAC\Binary\target2.exe       SUCCESS

2:43:44 AM     cmd.exe:704    IRP_MJ_CLOSE   D:\Giac\GIAC\Binary\target2.exe       SUCCESS

2:43:44 AM     cmd.exe:704    IRP_MJ_CLEANUP D:\Giac\GIAC\Binary\target2.exe       SUCCESS

2:43:44 AM     cmd.exe:704    IRP_MJ_CLOSE   D:\Giac\GIAC\Binary\target2.exe       SUCCESS

2:43:44 AM     cmd.exe:704    FSCTL_IS_VOLUME_MOUNTED       C:\     SUCCESS
2:43:44 AM     cmd.exe:704    FSCTL_IS_VOLUME_MOUNTED       C:\     SUCCESS
2:43:44 AM     cmd.exe:704    IRP_MJ_CREATE  C:\     SUCCESS Attributes: Any Options: Open

2:43:44 AM     cmd.exe:704    FASTIO_QUERY_BASIC_INFO       C:\     SUCCESS Attributes: D
2:43:44 AM     cmd.exe:704    IRP_MJ_CLEANUP C:\     SUCCESS
2:43:44 AM     cmd.exe:704    IRP_MJ_CLOSE   C:\     SUCCESS
2:43:44 AM     target2.exe:1212      IRP_MJ_CREATE  C:\     SUCCESS Attributes: Any
Options: Open Directory
2:43:44 AM     target2.exe:1212      IRP_MJ_CREATE  D:\Giac\GIAC\Binary\WS2_32.dll
       FILE NOT FOUND Attributes: Any Options: Open
2:43:44 AM     target2.exe:1212      FSCTL_IS_VOLUME_MOUNTED       C:\     SUCCESS
2:43:44 AM     target2.exe:1212      IRP_MJ_CREATE  C:\WS2_32.dll  FILE NOT FOUND
       Attributes: Any Options: Open
2:43:44 AM     target2.exe:1212      FSCTL_IS_VOLUME_MOUNTED       C:\     SUCCESS
2:43:44 AM     target2.exe:1212      IRP_MJ_CREATE  C:\WINNT\System32\WS2_32.dll  SUCCESS
       Attributes: Any Options: Open
2:43:44 AM     target2.exe:1212      FASTIO_QUERY_BASIC_INFO
       C:\WINNT\System32\WS2_32.dll  SUCCESS Attributes: A
2:43:44 AM     target2.exe:1212      IRP_MJ_CLEANUP C:\WINNT\System32\WS2_32.dll  SUCCESS

2:43:44 AM     target2.exe:1212      IRP_MJ_CLOSE   C:\WINNT\System32\WS2_32.dll  SUCCESS

2:43:44 AM     target2.exe:1212      FSCTL_IS_VOLUME_MOUNTED       C:\     SUCCESS
2:43:44 AM     target2.exe:1212      FSCTL_IS_VOLUME_MOUNTED       C:\     SUCCESS
2:43:44 AM     target2.exe:1212      IRP_MJ_CREATE  C:\WINNT\System32\WS2_32.dll  SUCCESS
       Attributes: Any Options: Open
2:43:44 AM     target2.exe:1212      IRP_MJ_CREATE  C:\WINNT\System32\WS2_32.dll  SUCCESS
       Attributes: N Options: Open
2:43:44 AM     target2.exe:1212      IRP_MJ_QUERY_INFORMATION
       C:\WINNT\System32\WS2_32.dll  SUCCESS FileInternalInformation
```

```
2:43:44 AM     target2.exe:1212      IRP_MJ_CLEANUP C:\WINNT\System32\WS2_32.dll  SUCCESS

2:43:44 AM     target2.exe:1212      IRP_MJ_CLOSE   C:\WINNT\System32\WS2_32.dll  SUCCESS

2:43:44 AM     target2.exe:1212      IRP_MJ_CREATE  C:\WINNT\System32\WS2_32.dll  SUCCESS
       Attributes: N Options: Open
2:43:44 AM     target2.exe:1212      FASTIO_QUERY_STANDARD_INFO
       C:\WINNT\System32\WS2_32.dll  SUCCESS Size: 69392
2:43:44 AM     target2.exe:1212      IRP_MJ_CLEANUP C:\WINNT\System32\WS2_32.dll  SUCCESS

2:43:44 AM     target2.exe:1212      IRP_MJ_CLOSE   C:\WINNT\System32\WS2_32.dll  SUCCESS

2:43:44 AM     target2.exe:1212      IRP_MJ_CLEANUP C:\WINNT\System32\WS2_32.dll  SUCCESS

2:43:44 AM     target2.exe:1212      IRP_MJ_CLOSE   C:\WINNT\System32\WS2_32.dll  SUCCESS

2:43:44 AM     target2.exe:1212      IRP_MJ_CREATE  D:\Giac\GIAC\Binary\WS2HELP.DLL
       FILE NOT FOUND Attributes: Any Options: Open
2:43:44 AM     target2.exe:1212      FSCTL_IS_VOLUME_MOUNTED       C:\     SUCCESS
2:43:44 AM     target2.exe:1212      IRP_MJ_CREATE  C:\WS2HELP.DLL FILE NOT FOUND
       Attributes: Any Options: Open
2:43:44 AM     target2.exe:1212      FSCTL_IS_VOLUME_MOUNTED       C:\     SUCCESS
2:43:44 AM     target2.exe:1212      IRP_MJ_CREATE  C:\WINNT\System32\WS2HELP.DLL SUCCESS
       Attributes: Any Options: Open
2:43:44 AM     target2.exe:1212      FASTIO_QUERY_BASIC_INFO
       C:\WINNT\System32\WS2HELP.DLL SUCCESS Attributes: A
2:43:44 AM     target2.exe:1212      IRP_MJ_CLEANUP C:\WINNT\System32\WS2HELP.DLL SUCCESS

2:43:44 AM     target2.exe:1212      IRP_MJ_CLOSE   C:\WINNT\System32\WS2HELP.DLL SUCCESS

2:43:44 AM     target2.exe:1212      FSCTL_IS_VOLUME_MOUNTED       C:\     SUCCESS
2:43:44 AM     target2.exe:1212      FSCTL_IS_VOLUME_MOUNTED       C:\     SUCCESS
2:43:44 AM     target2.exe:1212      IRP_MJ_CREATE  C:\WINNT\System32\WS2HELP.DLL SUCCESS
       Attributes: Any Options: Open
2:43:44 AM     target2.exe:1212      IRP_MJ_CREATE  C:\WINNT\System32\WS2HELP.DLL SUCCESS
       Attributes: N Options: Open
2:43:44 AM     target2.exe:1212      IRP_MJ_QUERY_INFORMATION
       C:\WINNT\System32\WS2HELP.DLL SUCCESS FileInternalInformation
2:43:44 AM     target2.exe:1212      IRP_MJ_CLEANUP C:\WINNT\System32\WS2HELP.DLL SUCCESS

2:43:44 AM     target2.exe:1212      IRP_MJ_CLOSE   C:\WINNT\System32\WS2HELP.DLL SUCCESS

2:43:44 AM     target2.exe:1212      IRP_MJ_CREATE  C:\WINNT\System32\WS2HELP.DLL SUCCESS
       Attributes: N Options: Open
2:43:44 AM     target2.exe:1212      FASTIO_QUERY_STANDARD_INFO
       C:\WINNT\System32\WS2HELP.DLL SUCCESS Size: 18192
2:43:44 AM     target2.exe:1212      IRP_MJ_CLEANUP C:\WINNT\System32\WS2HELP.DLL SUCCESS

2:43:44 AM     target2.exe:1212      IRP_MJ_CLOSE   C:\WINNT\System32\WS2HELP.DLL SUCCESS

2:43:44 AM     target2.exe:1212      IRP_MJ_CLEANUP C:\WINNT\System32\WS2HELP.DLL SUCCESS

2:43:44 AM     target2.exe:1212      IRP_MJ_CLOSE   C:\WINNT\System32\WS2HELP.DLL SUCCESS

2:43:44 AM     target2.exe:1212      IRP_MJ_CREATE  D:\Giac\GIAC\Binary\MFC42.DLL FILE
NOT FOUND      Attributes: Any Options: Open
2:43:44 AM     target2.exe:1212      FSCTL_IS_VOLUME_MOUNTED       C:\     SUCCESS
2:43:44 AM     target2.exe:1212      IRP_MJ_CREATE  C:\MFC42.DLL   FILE NOT FOUND
       Attributes: Any Options: Open
2:43:44 AM     target2.exe:1212      FSCTL_IS_VOLUME_MOUNTED       C:\     SUCCESS
2:43:44 AM     target2.exe:1212      IRP_MJ_CREATE  C:\WINNT\System32\MFC42.DLL   SUCCESS
       Attributes: Any Options: Open
2:43:44 AM     target2.exe:1212      FASTIO_QUERY_BASIC_INFO
       C:\WINNT\System32\MFC42.DLL   SUCCESS Attributes: A
2:43:44 AM     target2.exe:1212      IRP_MJ_CLEANUP C:\WINNT\System32\MFC42.DLL   SUCCESS

2:43:44 AM     target2.exe:1212      IRP_MJ_CLOSE   C:\WINNT\System32\MFC42.DLL   SUCCESS

2:43:44 AM     target2.exe:1212      FSCTL_IS_VOLUME_MOUNTED       C:\     SUCCESS
2:43:44 AM     target2.exe:1212      FSCTL_IS_VOLUME_MOUNTED       C:\     SUCCESS
```

```
2:43:44 AM     target2.exe:1212       IRP_MJ_CREATE  C:\WINNT\System32\MFC42.DLL   SUCCESS
       Attributes: Any Options: Open
2:43:44 AM     target2.exe:1212       IRP_MJ_CREATE  C:\WINNT\System32\MFC42.DLL   SUCCESS
       Attributes: N Options: Open
2:43:44 AM     target2.exe:1212       IRP_MJ_QUERY_INFORMATION
       C:\WINNT\System32\MFC42.DLL   SUCCESS FileInternalInformation
2:43:44 AM     target2.exe:1212       IRP_MJ_CLEANUP C:\WINNT\System32\MFC42.DLL   SUCCESS

2:43:44 AM     target2.exe:1212       IRP_MJ_CLOSE   C:\WINNT\System32\MFC42.DLL   SUCCESS

2:43:44 AM     target2.exe:1212       IRP_MJ_CREATE  C:\WINNT\System32\MFC42.DLL   SUCCESS
       Attributes: N Options: Open
2:43:44 AM     target2.exe:1212       FASTIO_QUERY_STANDARD_INFO
       C:\WINNT\System32\MFC42.DLL   SUCCESS Size: 995383
2:43:44 AM     target2.exe:1212       IRP_MJ_CLEANUP C:\WINNT\System32\MFC42.DLL   SUCCESS

2:43:44 AM     target2.exe:1212       IRP_MJ_CLOSE   C:\WINNT\System32\MFC42.DLL   SUCCESS

2:43:44 AM     target2.exe:1212       IRP_MJ_CLEANUP C:\WINNT\System32\MFC42.DLL   SUCCESS

2:43:44 AM     target2.exe:1212       IRP_MJ_CLOSE   C:\WINNT\System32\MFC42.DLL   SUCCESS

2:43:44 AM     target2.exe:1212       IRP_MJ_CREATE  D:\Giac\GIAC\Binary\MSVCP60.dll
       FILE NOT FOUND Attributes: Any Options: Open
2:43:44 AM     target2.exe:1212       FSCTL_IS_VOLUME_MOUNTED       C:\     SUCCESS
2:43:44 AM     target2.exe:1212       IRP_MJ_CREATE  C:\MSVCP60.dll FILE NOT FOUND
       Attributes: Any Options: Open
2:43:44 AM     target2.exe:1212       FSCTL_IS_VOLUME_MOUNTED       C:\     SUCCESS
2:43:44 AM     target2.exe:1212       IRP_MJ_CREATE  C:\WINNT\System32\MSVCP60.dll SUCCESS
       Attributes: Any Options: Open
2:43:44 AM     target2.exe:1212       FASTIO_QUERY_BASIC_INFO
       C:\WINNT\System32\MSVCP60.dll SUCCESS Attributes: A
2:43:44 AM     target2.exe:1212       IRP_MJ_CLEANUP C:\WINNT\System32\MSVCP60.dll SUCCESS

2:43:44 AM     target2.exe:1212       IRP_MJ_CLOSE   C:\WINNT\System32\MSVCP60.dll SUCCESS

2:43:44 AM     target2.exe:1212       FSCTL_IS_VOLUME_MOUNTED       C:\     SUCCESS
2:43:44 AM     target2.exe:1212       FSCTL_IS_VOLUME_MOUNTED       C:\     SUCCESS
2:43:44 AM     target2.exe:1212       IRP_MJ_CREATE  C:\WINNT\System32\MSVCP60.dll SUCCESS
       Attributes: Any Options: Open
2:43:44 AM     target2.exe:1212       IRP_MJ_CREATE  C:\WINNT\System32\MSVCP60.dll SUCCESS
       Attributes: N Options: Open
2:43:44 AM     target2.exe:1212       IRP_MJ_QUERY_INFORMATION
       C:\WINNT\System32\MSVCP60.dll SUCCESS FileInternalInformation
2:43:44 AM     target2.exe:1212       IRP_MJ_CLEANUP C:\WINNT\System32\MSVCP60.dll SUCCESS

2:43:44 AM     target2.exe:1212       IRP_MJ_CLOSE   C:\WINNT\System32\MSVCP60.dll SUCCESS

2:43:44 AM     target2.exe:1212       IRP_MJ_CREATE  C:\WINNT\System32\MSVCP60.dll SUCCESS
       Attributes: N Options: Open
2:43:44 AM     target2.exe:1212       FASTIO_QUERY_STANDARD_INFO
       C:\WINNT\System32\MSVCP60.dll SUCCESS Size: 401462
2:43:44 AM     target2.exe:1212       IRP_MJ_CLEANUP C:\WINNT\System32\MSVCP60.dll SUCCESS

2:43:44 AM     target2.exe:1212       IRP_MJ_CLOSE   C:\WINNT\System32\MSVCP60.dll SUCCESS

2:43:44 AM     target2.exe:1212       IRP_MJ_CLEANUP C:\WINNT\System32\MSVCP60.dll SUCCESS

2:43:44 AM     target2.exe:1212       IRP_MJ_CLOSE   C:\WINNT\System32\MSVCP60.dll SUCCESS

2:43:44 AM     target2.exe:1212       IRP_MJ_CREATE  D:\Giac\GIAC\Binary\target2.exe.Local
       FILE NOT FOUND Attributes: Any Options: Open
2:43:44 AM     System:8       IRP_MJ_CLOSE   C:\WINNT\system32\msclus.dll  SUCCESS
2:43:44 AM     System:8       IRP_MJ_CLOSE   C:\WINNT\system32\tdc.ocx     SUCCESS
2:43:44 AM     System:8       IRP_MJ_CLOSE   C:\WINNT\system32\mscpxl32.dLL       SUCCESS

2:43:44 AM     target2.exe:1212       FSCTL_IS_VOLUME_MOUNTED       C:\     SUCCESS
2:43:44 AM     target2.exe:1212       IRP_MJ_CREATE  C:\WINNT\System32\MFC42LOC.DLL
       FILE NOT FOUND Attributes: Any Options: Open
2:43:44 AM     target2.exe:1212       FSCTL_IS_VOLUME_MOUNTED       C:\     SUCCESS
```

```
2:43:44 AM     target2.exe:1212       IRP_MJ_CREATE  C:\WINNT\System32\MFC42LOC.DLL
       FILE NOT FOUND Attributes: Any Options: Open
```

The DLL files accessed, according to filemon, include: WS2_32.dll, WS2HELP.DLL, MFC42.DLL, MSVCP60.dll, and MFC42LOC.DLL.

19. The sniffer program indicates target2 is not attempting to access the network with the information collected. Another theory may be that the program is looking for a particular configuration to attack. If that is the case, the reason it appears to do nothing may be attributed to the fact that it did not find what it was looking for in the registry.
20. Selected Save in the Tokenmon window and saved the log to d:\during\tokenmonlog.txt. Here is an excerpt containing the references to target2.exe:

```
130     28.51457266    Explorer.EXE:1008      1016   ADJUST PRIVILEGES      00008F4E:
\\DSLAB01\Administrator       ENABLED: UNDOCK
131     30.82944452    Explorer.EXE:1008      1024   ADJUST PRIVILEGES      00008F4E:
\\DSLAB01\Administrator       ENABLED: INC_BASE_PRIORITY
132     30.83152188    Explorer.EXE:1008      1024   ADJUST PRIVILEGES      00008F4E:
\\DSLAB01\Administrator       DISABLED: INC_BASE_PRIORITY
133     30.85485334    cmd.exe:1008   1024    CREATE PROCESS 00008F4E:
\\DSLAB01\Administrator       Parent: Explorer.EXE:1008
134     30.85970200    csrss.exe:160  564     REVERTTOSELF   000003E7: \\NT
AUTHORITY\SYSTEM
135     30.86220232    csrss.exe:160  564     REVERTTOSELF   000003E7: \\NT
AUTHORITY\SYSTEM
136     30.86350919    csrss.exe:160  564     REVERTTOSELF   000003E7: \\NT
AUTHORITY\SYSTEM
137     48.22401423    target2.exe:276        280    CREATE PROCESS 00008F4E:
\\DSLAB01\Administrator       Parent: cmd.exe:276
138     63.24056186    target2.exe:264        1152   EXIT PROCESS   00008F4E:
\\DSLAB01\Administrator
139     67.38372223    services.exe:208       300    IMPERSONATE CLIENT OF PIPE
        000003E7: \\NT AUTHORITY\SYSTEM       000003E7: \\NT AUTHORITY\SYSTEM
140     67.38381051    services.exe:208       300    REVERTTOSELF   000003E7: \\NT
AUTHORITY\SYSTEM
141     67.38407507    services.exe:208       300    IMPERSONATE CLIENT OF PIPE
        000003E7: \\NT AUTHORITY\SYSTEM       000003E7: \\NT AUTHORITY\SYSTEM
142     67.38413681    services.exe:208       300    REVERTTOSELF   000003E7: \\NT
AUTHORITY\SYSTEM
143     67.38452485    services.exe:208       300    ADJUST PRIVILEGES      000003E7: \\NT
AUTHORITY\SYSTEM       ENABLED: AUDIT
```

21. Note that there are only two references to target2 – line 137 “create process,” and line 138 – “exit process.” No clues here as to what the program is doing. This log notes information about permissions/authority used for processes. The only thing we have learned, then, is that the target2 process executed with administrator privileges. That is not surprising since the user ID I was using was administrator equivalent.
22. Looked at the TDIMon window and noted no references to target2.exe. Did not save the results as they were not significant.
23. Selected the Portmon window and noted that there were no attempts to access either the serial or parallel printer ports. Closed without saving.
24. Did a file compare of the before and after directories to see if any changes were made in dll files or in the system directory. Used the following commands while in the d:\ directory:
    fc before\dlldir.txt after\dlldir.txt

fc before\systemdir.txt after\systemdir.txt. No differences were encountered except for the total space left on the drive as would be expected if nothing changed. This action would compare file names, file sizes, file dates, etc. – all information normally seen in the results of a dir command. There were no new file names, no names missing, and file sizes and dates last modified all remained the same.

25. Opened DumpReg and sorted the three registry hives by key. Then saved in tab delimited file in the d:\after\hivenamedump.txt files. That way I created 3 files – one each for the local machine hive, the users hive and the current user hive. These text files can be compared to the three I created in the d:\before directory to see if the registry changed.
26. Began a comparison process of files created during the test. The following files showed no change except for date and time the program ran that created the files: DumpACL of Policies, Users, Shares, Registry Local Machine Permissions, Registry Users Permissions, Groups, and Rights.
27. The following files did show differences: DumpACL of Services showed that the Kernel Wave Audio Mixer (kmixer) started running after the malware ran.
28. Checked the registry for Run and Runonce commands. Found under HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\OptionalComponents\ IMAIL, MAPI, and MSFS. Each of these indicates a key of Installed with a value of 1. Web research showed these entries to be normal. Comparison with a second computer in our office revealed the same settings.
29. Ran nmap a second time to compare results with earlier scan. Results were identical. Same ports were open. Nothing had changed.

There is no way to determine from the binary when the last time was that it was used. That being said, the create and modify times match, 2/20/03 at 12:45 PM. It appears the file was probably compiled at that date and time. If it was compiled on the compromised computer, that would indicate when the file was first available for use. Unfortunately, unzipping the file causes it to reflect a last accessed time and date to match the time and date it was unzipped.

Forensic Details

A string search of the drive image for appropriate strings from the above key word list will certainly indicate the presence of this code either compiled or uncompiled. That would provide the "largest" footprint. If the process is running, there would be an opportunity to discover it in memory using the task manager or pslist command(s). If the spelling of the binary is known or suspected, use the Windows search command by clicking on Start and Search. In the file title, type in the suspected title and click on Search Now. Even without that, one could use the same search command and specify a text string from the file to see if Windows finds it.

A DOS command can also be used in any given directory. Using the Find command: `C:\GIAC>find /i "creare" *` would find any file in the specified directory with the word “creare” in it. In this instance, I purposely executed the command in the directory where the executable existed and it pulled out the entire line where the word appeared as follows:

```
---------- TARGET2.EXE
impossibile creare raw ICMP socket
```

Using an updated version of dd.exe, (located at this URL: http://users.erols.com/gmgarner/forensics/), capable of dumping physical memory on a Windows 2000 computer, I created a file called physicalmemory.img using this command:

```
dd if=\\.\PhysicalMemory of=d:\giac\physicalmemory.img
```

This process took a while, since the memory of the loaner computer was 640 megabytes. Then I used the strings.exe command to create a text file of the strings in memory. That reduced the file size to 11 meg. Find was used to see if the string “creare” could be located in the memory image. The first run was conducted without having run target2.exe since the last reboot. This was to see if the program had covertly set itself up to run automatically on bootup. The first effort was not successful, in that it did not locate the search string. To look for a footprint in memory, I then ran target2.exe, dumped memory, created a text file, and did another string search, anticipating that the program would probably still reside somewhere in memory, even if it was no longer functional. Worst case scenario, it could be a TSR that is ready to be called from memory at any time based on some trigger. (Ex., incoming traffic on certain ports in a certain order, etc.) There has been no evidence, yet, of that being the case, however.

The second run of strings and find, after the post target2.exe dump of memory, also reported nothing of significance. The only strings that were found were the command lines entered while searching for the word “creare.” So, assuming a resident program would load the word “creare” into memory, it appears that target2.exe cleared itself out of memory after running.

Files accessed by target2.exe during runtime include the registry, as noted, as well as the aforementioned DLLs.

There was, as noted above, an IP address located in the code. Even more interesting, the ip was listed in UNC format with C$ at the end:

```
\\199.107.97.191\C$
```

I assumed from this string that the program may try to communicate with a particular computer at that address. C$ is the default designation for the system share of the C: drive on a Windows computer. Armed with that knowledge, I did a trace route to discover more information about that address. Using the command tracert 199.107.97.191 the first line resulting from the command said: “Tracing route to sbm191.dtc.apu.edu [199.107.97.191]”

After further research on whois, I received the following information:

```
Domain Name: APU.EDU
Registrant:
    Azusa Pacific University
    PO Box 7000
    Azusa, CA 91702-7000
```

```
   UNITED STATES
Contacts:
   Administrative Contact:
   John Reynolds
   Chief Information Officer
   Azusa Pacific University
   PO Box 7000
   Azusa, CA 91702-7000
   UNITED STATES
   (626) 969-3434
   jreynolds@apu.edu
   Technical Contact:
   James Stoker
   Network Administrator
   Azusa Pacific University
   PO Box 7000
   Azusa, CA 91702-7000
   UNITED STATES
   (626) 969-3434
   jstoker@apu.edu
Name Servers:
   NS.APU.EDU                  199.184.237.168
   CBRU.BR.NS.ELS-GMS.ATT.NET
   CMTU.MT.NS.ELS-GMS.ATT.NET
Domain record activated:    03-May-1994
Domain record last updated: 13-Aug-2002
```

It is logical to assume that either the program was created by a student at the above institution, or, more likely, that some unwitting student or employee of the institution, whose ID was most likely SBM191, may have had their computer compromised. If this assumption is correct, then their computer was being used by a hacker located somewhere else. Cooperation with the institution would be necessary to find out more about the identity of the computer and its administrator/user beyond these assumptions. More research would be necessary if this were a complete investigation.

Program Identification

The source code has not been located and could not, therefore, be compiled or compared. As mentioned earlier, source code with the foreign phrase was located, but it did not exactly match the information revealed by bintext.

Legal Implications

There is no way to prove that the program was executed on the system from which the binary was removed, since we were not provided with the disk image, logs, or other information that might provide that indication.

From a legal standpoint, were a hacker to install target2 on a system but not use it for any malicious activities, it would be difficult, if not impossible, to convict the hacker. Current law addresses loss in excess of $5k, damage leading to physical harm, intent to do harm – in short, things that would be hard to prove if the user did nothing malicious. On the other hand, if the user created a portal by hacking the box and loading target2, then used that portal to ultimately access personal data or gave other indications of an intent to do harm, the user could be liable and could serve up to 20 years, per the following statement acquired from CCIPS (Computer Crime and Intellectual Property Section) at the following site: http://www.cybercrime.gov/PatriotAct.htm.

> "Section 814 makes a number of changes to improve 18 U.S.C. § 1030, the Computer Fraud and Abuse Act. This section increases penalties for hackers who damage protected computers (from a maximum of 10 years to a maximum of 20 years); clarifies the *mens rea* required for such offenses to make explicit that a hacker need only intend damage, not a particular *type* of damage; adds a new offense for damaging computers used for national security or criminal justice; expands the coverage of the statute to include computers in foreign countries so long as there is an effect on U.S. interstate or foreign commerce; counts state convictions as "prior offenses" for purpose of recidivist sentencing enhancements; and allows losses to several computers from a hacker's course of conduct to be aggregated for purposes of meeting the $5,000 jurisdictional threshold."

And…

> "In United States v. Middleton, 231 F.3d 1207, 1210-11 (9th Cir. 2000), the court held that the definition of loss includes a wide range of harms typically suffered by the victims of computer crimes, including costs of responding to the offense, conducting a damage assessment, restoring the system and data to their condition prior to the offense, and any lost revenue or costs incurred because of interruption of service.
> *Amendments:* Amendments in Section 814 codify the appropriately broad definition of loss adopted in Middleton. 18 U.S.C. § 1030(e)(11)."

In addition to legal restrictions on hackers, our corporation's internal policies certainly prohibit such activities. Any user would be in violation of our policies, were he or she to install target2 or any other malware on any computer in our network. The policy addresses the principle of least privilege limiting access to data based on job function, defines violations of the policy, and threatens the user with discipline and/or dismissal as a result of such violations.

Here are some excerpts from our policy which directly or indirectly address this issue:

> "Violations
>
> "Violations of this data security policy may include, but are not limited to, any act that:
>
> - Exposes the corporation to actual or potential monetary loss through the compromise of data security.
> - Involves the disclosure of trade secrets or confidential information or the unauthorized use of corporate data.
> - Involves the use of data for illicit purposes, which may include violation of any law, regulation or reporting requirement of any law enforcement or government body.
>
> "Classified Data - This information may only be viewed by persons specifically identified by the owner of the data. Access to this data must be requested by the employee's cost center manager or designee and authorized by the data owner. Read and update capabilities must be

specifically defined. Where appropriate, on-line mechanized access should be restricted by program/function control to those persons cleared by the owner for access. Any eye-readable documents (paper, microfiche) should only be handled by people cleared to route restricted data. Interim storage shall be secured. Release of the document will only be to authorized individuals. Information sent to a remote site shall be placed in an individual tamper proof package. It should be addressed to the recipient and marked confidential. Permission to copy this information for dissemination in or outside the corporation must be granted by the owner of the data. All requests must be in writing. Information that will become part of a lawsuit or criminal investigation must not be released unless the request is in writing from the proper authorities. Mechanized transfer of data from one source to another shall not take place unless comparable protection can be provided at the transfer site. Access to this information by contractors or vendors must be covered by a non-disclosure document signed by the vendor company or the individual.

"Business Confidential - This information may only been viewed by persons who have a need to know based on their job function. Access to this data must be approved by a cost center manager or designee and the data owner. Read and update capabilities must be specifically defined. Permission to copy the information for dissemination in or out of the corporation must be approved by the owner of the data. Verbal communication of the information shall always be cleared with the owner. Information that will become part of a law suit or criminal investigation must not be released unless the request is in writing from the proper authorities. Data in eye-readable form should be stored in a way that unauthorized persons will not have easy access. After normal business hours, this information should be locked up. Access to this information by contractors or vendors must be covered by a non-disclosure document signed by the vendor company or the individual.

"Client - This information is owned by clients and maintained via contract by [company]. This information may only be viewed by persons who have a need to know based on their job function. External access (clients/vendors, etc.) to this data is controlled by the client. [company] read and update capabilities must be specifically defined and approved by a business unit manager or the Operations Manager (owner of Production files). Access to this information by contractors or vendors must be covered by a non-disclosure document signed by the vendor company or the individual.

"Appropriate use/violations

"Appropriate use of company information should be governed by its practical business use and always with good judgment. Using any company information for personal gain is expressly forbidden.
"Violations of this policy can lead to disciplinary action up to and including termination."

Clearly, from the above policy statements, the corporation would be within rights to discipline and/or dismiss any employee acquiring or giving unauthorized access to data.

Without knowing exactly what the binary will do to a computer in a "destructive" way, it is difficult to assess what the legal ramifications would be, were we to discover target2.exe or its equivalent on one of our company's computers. There is a sense in which, even if it is benign as it appears from our research, that there is a significant cost in time and resources just to verify that it is benign. The whole problem goes back to intent. If the user intended even to imply that target2.exe would or could cause damage, regardless of the program's capability, then there may be grounds for litigation.

For example, should the malicious user manage to place the program on one of the corporation's internet facing applications and the application is then discovered by support personnel, a CERT (Computer Emergency Response Team) would automatically be called into action. The CERT is made up of an ad hoc group that could include at least one team member and one backup from several groups. The groups represented would include Implementation, Corporate Security, Testing, QA, Benchmark, and even members of senior management. Other could be called into the process as needed.

The costs to the corporation would include the hours committed by each of these persons, the cost of delays of normal operations, testing, and implementation caused by diversion of these personnel, and costs of diversion of equipment necessary for testing and researching the issue. If the personnel are working the problem around the clock, there would be the cost of overtime. If sensitive data is compromised, there is the cost of recovering good data from backups or by re-keying, as well as the cost to our company's reputation, which could be astronomical. When all of these costs are added together, even in the case of an executable that turns out to be benign, they can easily exceed the $5,000 requirement of the law.

One of the most frustrating parts of this process is determining costs for something that has no absolute monetary value. If the compromise becomes public and the corporation affected happens to be an ASP (Application Service Provider), their reputation as a secure repository for data could be destroyed. This, in turn, could literally put the company out of business. The true value of that reputation would then equate to the business lost. Even that may remain an unknown, since customers are not always open about their reasons for taking business elsewhere.

So the legal implications are many, expensive, and ongoing for both the corporation and the perpetrator of the event. The damage can be extensive to countless persons for many years to come, even in the event that the malware is considered benign. In the case where a program is destructive to hardware, software, and/or data, the costs could be immeasurable.

Interview Questions

A few questions could be used to "break the ice" and get the malware user talking. Here are some examples that might help:

- How would it be possible to access a computer using the ICMP (ping) protocol? (May be willing to share his knowledge if he doesn't suspect that you know what you know.)
- What steps do you have to go through to compile C code? (To determine knowledge of C programming.)
- If above is yes, what is your current experience with C coding? Have you written any code outside of the company? What kind of code have you

written? (Some can be verified with management. This will appeal to his/her ego – to be able to share all the wonderful things they have written and, of course, how smart they are.)

- How do you have access to [the system that was compromised.] Who provided authorization for you to have access to the system? Why do you need that access? (Verify with that individual if name is given)
- Can you tell me any web sites where I could find programs that I could use to get into a system I am having trouble accessing? Are you familiar with Phrack.org? What about attrition.org? sourceforge.net? packetstorm.linuxsecurity.com? (Can be verified by firewall logs as to whether or not he/she has accessed sites where code is available or where defaced sites are displayed.)

These could lead to other questions that may lend even more help to the investigation.

Additional Information

Outside sources used in this discourse on target2.exe, ICMP, and/or the legal ramifications, include the following:

Computer Crime and Intellectual Property Section (CCIPS). “Field Guidance on New Authorities That Relate to Computer Crime and Electronic Evidence Enacted in the USA Patriot Act of 2001.” 5 November 2001. URL: http://www.cybercrime.gov/PatriotAct.htm (3 April 2003).

Dashie. “BFi numero 7, anno 2 - 25/12/1999 - file 13 di 22.” 25 December 1999. URL: http://www.s0ftpj.org/bfi/online/bfi7/bfi07-13.html. (23 May 2003).

Garner, Jr., George M. “Forensic Acquisition Utilities.” 25 April 2002. URL: http://users.erols.com/gmgarner/forensics/. (23 May 2003).

IANA. “Port Numbers.” 13May2003. URL: http://www.iana.org/assignments/port-numbers (16May2003)

Part 2 – Option 1: Perform Forensic Analysis on a system

Synopsis of Case Facts

To fulfill this requirement, the author acquired a hard drive from an NT system that is no longer being used. As the older computers slide into disuse due to the dated operating systems and software that run on them, they begin to collect in various cubicle junk piles until the cubicle police come to collect them. That, of course, won't happen for many months after the PCs begin appearing and collecting dust.

With appropriate permission, of course, I visited one of our junk piles where NT systems were said to exist. Because our site has a proliferation of FAT and NTFS drives, and because the earlier version of Autopsy provided in the SANS course would not work on FAT drives, I kept testing drives until I found one with an NTFS partition. I then recorded the appropriate information as if performing a Forensics Analysis on a suspect system for the purposes of this report.

So, in short, there is no case pending, nor should there be, in regard to the selected NTFS drive. As a result, I looked at the drive to see if I could find any information indicating misuse of company assets and time, as well as any sign of proprietary information that should have been erased from the disk before the computer was given away.

Describe the system(s) you will be analyzing

The system was acquired from a development department within our corporation. Since we received the system without knowing anything about it, analysis will have to let us know what type of development was performed on the computer. Suppositions will be based on the type of software loaded on the computer and any documents that might support those theories.

The system included an Ethernet card and was connected to the corporate network. The operating system is NT. The exact version information will have to be gleaned from analysis. Based on the age of the computer and company policies, it is probably NT 4.0. The hard drive was removed from the computer and tagged appropriately, as was the rest of the equipment. All equipment being held for evidence is stored in a secured cage in a locked closet. A log sheet is kept of those who access the equipment and the purpose for which the equipment might be removed from storage. This is to ensure that we have a solid chain of evidence should we have to take a case to court.

Hardware

The system acquired is a legacy Compaq computer labeled with the following numbers and evidence tag numbers:

- Evidence Tag: 01 ....Asset tag number 21632
- Evidence Tag: 02 ....Compaq Computer
  Model Number DP2000 5233MMX 3200/CDS DOM
  Serial Number 9752BNT2D175
  With an internal 3.5 inch floppy drive, internal CD Rom reader, internal 3.2 GB hard drive, internal sound card, and internal Network Interface Card (NIC).

- Evidence Tag: 03 . Compaq (Manufactured by Fujitsu) IDE 3.2 GB Hard Drive
  Compaq Part Number of drive: 165580-001
  Compaq Serial Number of drive: 06002305F52
  Fujitsu Model Number on drive: MPD3108AT
  Fujitsu Part Number on drive: CA05177-B815000C
  Fujitsu Serial Number on drive: 33084054
  Date of manufacture: 1999-10
- Not labeled .... Operating System Windows NT

In an actual case, the computer would have been photographed where discovered, and at every stage of dismantling, to be able to ensure no tampering with the equipment if asked in the courtroom. The pictures could be compared to the actual physical evidence to determine if any of it has been altered.

No monitor was seized as this computer was not currently in use and there was no monitor with it.

Image Media

The first step in the forensics procedure was to create a forensics computer. Because this is a multi-step process, and because it may be repeated a number of times for different investigations, I have included the steps required to set up such a computer. For the purpose, I acquired a Compaq EN model with a Pentium III 866 processor. The computer has 128 MB of memory, a 20 GB hard drive, integrated sound card, network, and video. (Model number: ENS/P866/20e/6/128cv US). This system already had Windows 2000 installed on it. The system was repartitioned for Linux and prepared by sterilization. The command dd if=/dev/zero of=/dev/hda1 bs=512 was used to fill the entire partition with zeroes. The same command was repeated for hda2, hda3, and hda5 because those partitions were for the linux system. The partition used by the system for diagnostics, /dev/hda4, was not overwritten. This was to make certain that the Compaq diagnostics remained available on the hard drive. From a support standpoint, it saves so much time when diagnosing hardware issues to have the diagnostics available. Speaking as a former hardware tech, I am trying to make life easier for the LAN/Desktop gurus, or myself, should it become necessary to work on and diagnose problems on this Compaq computer.

Linux 7.2 (kernel 2.4.7-10) was installed on the Compaq with all options included. Then the kernel was rebuilt with NTFS and FAT enabled so that those drives would be recognized by the system. This was important for our exercise as it enabled creating an image of the acquired drive. Without NTFS and FAT recognition, a Windows version of dd could have been used to create the image from a secondary drive, but I thought it best that multiple systems be readable by the same forensics computer for consistency.

The following forensics software was originally installed on the forensics computer from the SANS Forensics CD supplied to our forensics class:

- Ethereal version 0.9.7
- The Coroner's Toolkit version 1.00
- Task version 1.00
- Autopsy version 1.50

After attempting to read the acquired NTFS disk image with some difficulty, the author downloaded and installed new versions of:

- Task (1.60)
- Autopsy (1.70)

The new features of the later versions allowed access to FAT and NTFS images, and had other time saving features.

Here is a list of included software in the above installs and the purpose of these packages:

Ethereal	a sniffer and packet analyzer that "sniffs," or records packets of information being passed about on an Ethernet network. Because there are a number of conventional protocols. Protocols are ways of putting data into a format along with the destination for the data and other information that helps ensure the integrity of the data when it gets to its destination. The construction of protocols is fairly rigid. If the information is not sent in the right order, in other words, it can be rejected at the receiving end. Ethereal automatically demystifies this process to some extent, interpreting for the user what protocol is being used and what data resides in the information being sent.
The Coroner's Toolkit	a package of several programs including those following with a – in front of them. These programs are designed specifically for forensics purposes as implied by this name. Just like the city coroner's office examines a dead body to determine what happened to it, a computer forensics specialist may use tools like this to examine a "body" of data that used to represent a live system.
- MACTime	This tool combines features of ls and stat commands to allow correlation of Modify, Access, Create (MAC) date/times. It can operate off of a live system, or from a disk image file. Such correlation between data provided by different ls and stat commands would be time intensive and awkward. MACTime pulls the data together so that it can be easily presented to the user.
- Lazarus	"*lazarus* tries to revive things that have died and gone into the binary spirit world... deleted files, data in memory, swap, etc." - according to Dan Farmer in the on line man page located at: http://staff.washington.edu/dittrich/talks/blackhat/tct/man/man1/lazarus.1.html. Mr. Farmer goes on to explain that lazarus analyzes a block of data and, if it meets criteria that make it look suspiciously like a file, the block will be saved as a file. Lazarus is very time consuming to run.
- Grave Robber	A review of the perl script, (Perl is a programming language), indicates that grave-robber will run a number of other programs/processes in order to gather data and store it. Data gathered can include running process information and inode information. (See explanation of inodes below under icat section.) Data is stored in files including body, body.S, MD5_all,

	MD5_all.md5. Programs run by grave robber include: icat, MACTime, md5sum, pcat, system commands like ps (lists process information), lsof (lists open file information), netstat (lists network status information), and df (lists file system information).
- pcat	Process information gathering program that runs on live memory. User must be careful as this can crash processes and servers. Pcat is used by Grave Robber as noted above to collect, concatenate, and save process information. Processes are essentially programs that run simultaneously in the background in all operating systems. Malware may start, kill, or change running processes and thereby disrupt normal operations.
- icat	inode information gathering program that creates files (concatenates contents) using inode numbers. icat is also used by Grave Robber as noted above. Every UNIX file has an inode that holds information about the file. Every file's inode has an inode number. In UNIX systems, information such as ownership, access permissions, and file type are stored in the inode.
- ils	Lists inode information on a specified device. By default, ils lists only the inodes of removed files.
- unrm	Copies blocks of data. By default, unrm copies unallocated data blocks. Creates files out of "dead" space.
Task	"The @stake Sleuth Kit (TASK) is the only open source forensic toolkit for a complete analysis of Microsoft and UNIX file systems. TASK enables investigators to identify and recover evidence from images acquired during incident response or from live systems. TASK is also open source, allowing investigators to verify the actions of the tool or customize it to specific needs." Author: Daniel Veillard, URL: http://rpmfind.net/linux/RPM/cooker/contrib/i586/task-1.52-2mdk.i586.html.
Autopsy	A Graphical User Interface incorporating most of the above programs into one point and click environment. It makes many of the aspects of forensics a "breeze," compared to running individual commands and collecting the output, then organizing the output in a logical manner. All of this is performed automatically by Autopsy.

Here are the steps used to install the above software:

Ethereal (from SANS CD)

First, make sure GTK+, Perl, and libpcap are installed.

- whereis perl [should give a response indicating where perl is located]
- rpm –q libpcap [should provide current version of libpcap]
- rpm –q gtk+ [should provide current version of gtk+]

Now install the code

- create and go to /usr/local/src directory
- mount /mnt/cdrom

- cd /usr/local/src
- tar zxvf /mnt/cdrom/UnixForensics/ethereal/ethereal*.tar.gz
- cd /usr/local/src/ethereal-0.9.1
- ./configure
- make
- make install

The Coroner's Toolkit

- cd /usr/local/src
- zcat /mnt/cdrom/UnixForensics/tct-1.09.tar.gz | tar xvf –
- cd tct-1.09
- make
- edit coroner.cf to taste [not necessary to change – will function as is]

Task

- cd /usr/local/src
- download task-1.60.tar.gz from http://www.atstake.com/research/tools/task/ to the same directory location as above
- zcat task-1.60.tar.gz | tar xvf -
- ln -s /usr/local/src/task-1.60 /usr/local/task
- cd task-1.60
- make

Autopsy

- cd /usr/local/src
- download autopsy-1.70.tar.gz from http://www.atstake.com/research/tools/autopsy/ to the /usr/local/src directory.
- zcat autopsy-1.70.tar.gz | tar xvf -
- cd autopsy-1.70
- make
- during the make process, you will be asked for some parameters.
- If you have downloaded the NIST National Software Reference Library (NSRL), you can point to its path during the install.
- Choose a location for the "Locker" directory that has enough space on it for disk images.

Once the forensics computer was set up and operational, used drives were obtained for testing. The fstab file was checked to be certain it contained no references to /dev/hdb since that could cause the drive to automount. Since the forensics computer did not have an extra secondary IDE cable, I temporarily removed the IDE CDRom and plugged in the drive to be tested in its place. After installation of the hard disk, the system booted with the following message:

```
"The following configuration options were automatically updated:
Disk:  20020 MB                    WDC WD200BB-60AUA1
```

```
        Disk:    3249 MB                    FUJITSU MPD3032AT
        If you are running Unix, you need to configure your system using the Computer Setup
    Utility (F10)"
F1:  Save Changes
```

After pressing F1 to save the changes and continue, the system rebooted and run Grub. I selected Linux with NTFS support. Note that the entire Fujitsu hard drive, the suspect disk, is about 3.2 GB in size.

Running fdisk –l gives the user a listing of the current partition information. Now that the evidence drive has been installed as a secondary drive, fdisk –l displayed the following in addition to the partition information for /dev/hda/:

```
Disk /dev/hdc:  128 heads,  63 sectors,  787 cylinders
Units = cylinders of 8064  *  512 bytes

        Device Boot          Start    End          Blocks       Id      System
/dev/hdc1          *             1    509        2052256+        6      FAT16
/dev/hdc2                      510    787         1120896        7      HPFS/NTFS
```

This indicates that there are two partitions on the drive. The first, hdc1, is a FAT16 partition and is the largest of the two. The second, hdc2, is HPFS/NTFS.

Once the drive was connected and recognized, the next step was to create an md5 hash of the hard drive partitions themselves. This information can then be retained to ensure there has been no tampering with the drive throughout the course of the investigation. These hashes will be compared with the actual partition images to ensure no tampering.

The following commands were used to create the hashes:

```
md5sum  /dev/hdc1  >  devhdc1.md5
md5sum  /dev/hdc2  >  devhdc2.md5
```

After changing to the subdirectory created for these images, I used a dd command to create an image of each of the partitions. Here are the commands used:

```
dd if=/dev/hdc1  of=image_021632_hdc1
dd if=/dev/hdc2  of=image_021632_hdc2_ntfs
```

Once these two files were created, a hash was run of each file for comparison to the original file. The following commands were used:

```
md5sum image_021632_hdc1 > image_021632_hdc1_md5sum
md5sum image_021632_hdc2_ntfs  > image_021632_hdc2_ntfs_md5sum
```

All of the sums created were concatenated into one file called md5.txt. The command used to create the file was cat devhdc1.md5 > md5.txt. Then each of the other 3 files were added by using the commands

```
cat image_021632_hdc1_md5sum >> md5.txt
cat devhdc2.md5 >> md5.txt
cat image_021632_hdc2_ntfs_md5sum >> md5.txt
```

Using two piping commands: >> allowed the new line(s) to be added to the file as opposed to replacing the file as would happen with only one >.

Here is the resulting file:

```
1e81c39052d2c4907f5d489c5a1eab22  /dev/hdc1

1e81c39052d2c4907f5d489c5a1eab22  image_021632_hdc1

65f7b07c1ce716e88be8b78fbb61b155  /dev/hdc2

65f7b07c1ce716e88be8b78fbb61b155  image_021632_hdc2_ntfs
```

Now that all of this information has been put into a file format, we next burned the following files onto a CDRom, making two copies. One to archive, and one for law enforcement, should they need a copy:

- devhdc1.md5
- devhdc2.md5
- image_021632_hdc1
- image_021632_hdc1_md5sum
- image_021632_hdc2_ntfs
- image_021632_hdc2_ntfs_md5sum
- md5.txt

The hard disk was removed after a shutdown, tagged with the evidence number, and stored in a secure place. In the case of our corporation, there is a secure "cage" area that has limited access and which is located in a secure closet. Now we are ready to perform an analysis of the media.

Media Analysis of System

We used Autopsy 1.70 to analyze the system. Run Autopsy with the following commands:

- cd /usr/local/src/autopsy-1.70
- ./autopsy 8888 localhost &
- You should be able to right click on the URL presented and run the browser from there. If not, just enter the same information in the browser address line that appears in the terminal screen after you run autopsy. (Don't close the window where you started autopsy!)

Autopsy is a browser app, so once the autopsy server is started, the user is presented with a GUI interface. To start a new case, click on New Case button.

- Enter the Case Name – this will be used to create a subdirectory off of the Locker directory you created.
- Enter a one line description
- Enter the names of investigator logins without spaces. Up to 10 investigators' logins can be associated with one case.
- You will be informed that the case has been created along with the directory and associate file(s)
- Click on OK

- Now you will be presented with the Case Gallery. If you have other cases in the system, they will appear here. Select the case you wish and click on OK.
- You will need to add a host in the Host Gallery. Click on Add Host.
- Enter the Host Name which will become a directory name as well.
- Enter a one line description of the host.
- Enter the time zone for this host. (This will be used for the investigation)
- Enter any time skew from your system clock + or – in seconds.
- Enter the path of the known bad files, if any.
- Enter the path of the Ignore hash database – good files to ignore, if any.
- Click on Add Host.
- You will be told the host has been added and what directory and files have been created.
- Click on OK, then in Host Gallery, click on OK again to go to Host Manager.
- This is where the actual work will take place. Click on Add Image. You will be informed as to where the image files reside. If there are files there, they will be listed. If not, add/move the file(s) to that directory.
- Click on Refresh to show the file on Add A New Image.
- Select the file you wish to add using the pull down.
- Identify the type of system (bsdi, fat, fat12, fat16, fat32, freebsd, linux-ext2, linux-ext3, ntfs, openbsd, or solaris)
- Identify the mounting point (ex. /mnt/hack/root)
- You can also provide an MD5 value so that the system can do an integrity check.
- Check Verify before adding image.
- Click on Add Image.

Once these steps have been taken, the files are waiting for your forensics activities. Open Autopsy, a case, a host, and select the image you wish to work with. To begin with, I selected the hdc1 FAT partition.

My first objective in viewing the partition was to determine if it had a system on it and what system it was. By clicking on the image details tab, the following information was displayed:

```
FILE SYSTEM INFORMATION
--------------------------------------------
File System Type: FAT
OEM: MSWIN4.1
Volume ID: 962336482
Volume Label: NO NAME
File System Type (super block): FAT16
```

This confirmed that the system was Microsoft Windows 4.1, according to the image detail signature. A quick look at the NTFS partition displayed:

```
FILE SYSTEM INFORMATION
--------------------------------------------
File System Type: NTFS
Volume Serial Number: 1CB36DDE1CB36DDE
Version: Windows NT

META-DATA INFORMATION
--------------------------------------------
```

```
Range: 0 - 4823
Root Directory: 5

CONTENT-DATA INFORMATION
-------------------------------------------
Sector Size: 512
Cluster Size: 2048
Total Cluster Range: 0 - 560446
```

Next I determined who the main user of the system was. In the profiles directory, there were several users listed as subdirectories. The earlier two had been deleted. It appears the computer was first used in March of 2000 since the earliest user directory was created then, as well as the Administrator user directory and the Default User directory. The user that was apparently active on the system began using it in June of 2000. My assumption is that the other two users with deleted profiles were technicians who had worked on the computer temporarily either before or while the main user was active on the system.

I checked the host file and found that there were two entries indicating that this user visited the mainframe host at our corporate site and also one of our client's hosts. The user had 3270 software and Delphi development components on his ntfs partition. Also, in the i386 subdirectory on the ntfs partition, it appeared that the user had attempted a reinstall of NT base files on the last day of the computer's use. Without further diagnosis, it would appear he was having problems with the computer, since the last attempted use was an NT re-install.

I selected the File Analysis tab and chose, first, to see what files had been deleted. There are several assumptions that can be made based on deleted files. If there are few or no deleted files on the system one may assume the suspect user could have had a reason to wipe the disk clean of certain information or clues. If there are many deleted files, a quick check may indicate whether this person had attempted to hide their tracks with a simple delete command.

As the number of deleted files was quite high, it was apparent that either the user did not know how to "wipe" the drive, or felt he had nothing to hide. As the user was a developer, as evidenced by the existence of development tools on his computer, he had a plethora of deleted temp files and other files that were obviously associated with his job tasks as well as the virus checker updates. In addition, as he obviously spent a great deal of time on the internet, there were many temp files in the Temporary Internet Files folder under content.ie5. These indicated locations visited, including mapquest and other ordinary sites, as well as a few sites not really necessary for work. See a discussion of this below.

I noted a good thing – his virus checker had been in use even the last day of use. The last accessed date was the last known date of operation for the key files in the Program Files\Navnt directory used by Norton Anti-virus.

Choosing a File Type sort on the FAT partition, where the user kept most of his documents, we searched for documents that could indicate inappropriate use of company assets. The documents noted all had to do with business – statements of work, functional specifications, detail design, and other documents one would expect to be associated with a developer's computer. (The most interesting files found under My Documents were wav files and mpgs, but even those were boringly benign. The typical startrekdoor.wav, doh.wav, and other essentials for any serious programmer's computer, as well as the new gold, silver and bronze awards that were distributed shortly after the most recent Olympics in gold.mpg, silver.mpg, and bronze.mpg.)

A search for known sniffer tools and other clandestine software not expected on a corporation computer, yielded no results. Filenames searched for include the Linux/UNIX tools that have been ported over to Windows, such as tcpdump (windump), ls, grep, etc. In addition, we looked for win-sniffer, sniffem, None of those tools exist on this drive indicating that this user was not involved in such activities as sniffing traffic on the corporate network or experimenting with downloaded tools. (All of this, of course, would be expressly forbidden by our Security Policies.)

I turned to the WINNT\Profiles\ and its subdirectories to look for information on recent files executed by the users as well as the internet activity. The following subdirectories were searched with the noted results:

- Administrator\Recent:

☑ Extra (used to access the Mainframe) was last accessed 8/28/02.

☑ Notes.ini.lnk (accessed when opening Lotus Notes) was last accessed the same day

- Administrator\Cookies:

☑ Index.dat was the only file and it indicated no internet activity
Administrator\Temporary Internet Files\Content.IE5\5vypxqfb and 8nn89m8t

☑ Files were created 9/21/2001 which corresponds to a date that all local PCs were updated with the latest service packs for IE5. These two directories appear, based on file names in them, to have been created by the service pack upgrades done at that time. There is evidence of access of the IE update site at Microsoft here. There is also evidence that the system, at that time, was Windows 98, not NT, because the download URL spelled out Windows 98/me for the service pack upgrade.

☑ Index.dat file in the Content.IE5 directory spelled out the URLs for the Service Pack upgrades.

- Lbxxxx2\desktop (user ID for the person using the computer being investigated.)

☑ Located a file called truck.txt that appeared to have two RACF passwords and a PIN stored in it. This verified the need to wipe this, as well as any other corporate computer, before allowing them to be discarded or given away. IT would be easy to provide a competitor with some highly valuable information residing on old corporate computers.

- Lbxxxx2\favorites (containing the following saved locations on internet)

☑ http://www.4x4parts.com (4x4 parts bulletin board specializing in Nissan)

☑ www.boltblue.com (a site with free e-mail and goodies for cell phones like ring tones)

☑ free.aol.com

☑ www.freebielist.com/phone.htm (a site with cell phone ring tones, etc.)

☑ djgremlin.mine.nu (unable to verify)

☑ home.iwon.com (a site that tracks where you go on the web and awards prizes based on various criteria, including links clicked at their site).

☑ Various University of North Florida sites – must be a fan.
Lbxxxx2\history\history.ie5 and subdirectories dated 8/26-8/28/2002

☑ Index.dat files from three directories. Using strings was able to extract the URLs visited by the user those three days. I used the export feature of Autopsy to send the file to /mnt/floppy/ieindex200208dd.txt where dd stood for the day of the file. Here is a sanitized excerpt from a single day's file for 8/26/02:

```
Client UrlCache MMF Ver 5.2
:2002082620020827: lbxxxx2@http://plus.iwon.com/aornum/aornum_IWPRM.html?src=IWPRM&u=...
:2002082620020827: lbxxxx2@:Host: plus.iwon.com
:2002082620020827: lbxxxx2@http://www.madamemercury.com
:2002082620020827:
lbxxxx2@http://pub175.ezboard.com/ftunfsfrm4.showMessage?topicID=225.topic
:2002082620020827: lbxxxx2@:Host: pub175.ezboard.com
:2002082620020827:
lbxxxx2@http://pub175.ezboard.com/ftunfsfrm18.showMessage?topicID=2460.topic
:2002082620020827:
lbxxxx2@http://pub175.ezboard.com/ftunfsfrm18.showMessage?topicID=3031.topic
:2002082620020827:
lbxxxx2@http://pub175.ezboard.com/ftunfsfrm18.showMessage?topicID=3026.topic
:2002082620020827:
lbxxxx2@http://pub175.ezboard.com/ftunfsfrm18.showMessage?topicID=3022.topic
:2002082620020827:
lbxxxx2@http://pub175.ezboard.com/ftunfsfrm18.showMessage?topicID=3015.topic
:2002082620020827:
lbxxxx2@http://pub175.ezboard.com/ftunfsfrm18.showMessage?topicID=3019.topic
:2002082620020827:
lbxxxx2@http://pub175.ezboard.com/ftunfsfrm18.showMessage?topicID=3018.topic
:2002082620020827:
lbxxxx2@http://pub175.ezboard.com/ftunfsfrm18.showMessage?topicID=3016.topic
:2002082620020827:
lbxxxx2@http://pub175.ezboard.com/ftunfsfrm18.showMessage?topicID=3030.topic
:2002082620020827: lbxxxx2@http://pub175.ezboard.com/btunfs
:2002082620020827: lbxxxx2@http://pub175.ezboard.com/ftunfsfrm18
:2002082620020827:
lbxxxx2@http://pub175.ezboard.com/ftunfsfrm4.showMessage?topicID=227.topic
:2002082620020827:
lbxxxx2@http://pub175.ezboard.com/ftunfsfrm4.showMessage?topicID=229.topic
:2002082620020827: lbxxxx2@http://pub175.ezboard.com/ftunfsfrm4
:2002082620020827:
lbxxxx2@http://www.iwon.com/home/prizes/pm3_overview/0,21311,,00.html?P...
:2002082620020827: lbxxxx2@:Host: home.iwon.com
:2002082620020827:
lbxxxx2@http://bfc.iwon.com/ad/pu3/index/id/iwon_POPUNDER_under.html?po...
:2002082620020827: lbxxxx2@:Host: bfc.iwon.com
:2002082620020827: lbxxxx2@http://news1.iwon.com/odd/article/id/191212|oddlyenough|08-26-
2...
:2002082620020827: lbxxxx2@:Host: my.iwon.com
:2002082620020827:
lbxxxx2@http://www.iwon.com/home/technology/tech_news/0,2108,technology...
:2002082620020827: lbxxxx2@:Host: www.iwon.com
:2002082620020827: lbxxxx2@http://my.iwon.com/index.jsp?PG=global?SEC=bnav
:2002082620020827: lbxxxx2@:Host: news1.iwon.com
:2002082620020827: lbxxxx2@http://news1.iwon.com/odd/article/id/262268|oddlyenough|08-26-
2002::10:10|reuters.html
:2002082620020827:
lbxxxx2@http://cache.unicast.com/upload/nxtl/q3fe/nxtl_q3fetch_a_100_Iwon_11057_html_ad_d
oc.html
:2002082620020827: lbxxxx2@:Host: cache.unicast.com
:2002082620020827: lbxxxx2@http://news1.iwon.com/odd/article/id/50708|oddlyenough|08-26-
2002::09:41|reuters.html
:2002082620020827: lbxxxx2@http://news1.iwon.com/odd/article/id/46352|oddlyenough|08-26-
2002::09:41|reuters.html
```

```
:2002082620020827:
lbxxxx2@http://ad.trafficmp.com/tmpad/content/reliaquote/rq_ad_720x300.html
:2002082620020827: lbxxxx2@http://news1.iwon.com/odd/article/id/262258|oddlyenough|08-26-
2002::09:23|reuters.html
:2002082620020827: lbxxxx2@:Host: ad.trafficmp.com
:2002082620020827: lbxxxx2@http://news1.iwon.com/odd/article/id/262262|oddlyenough|08-26-
2002::09:28|reuters.html
:2002082620020827: lbxxxx2@http://news1.iwon.com/odd/article/id/39680|oddlyenough|08-26-
2002::09:20|reuters.html
:2002082620020827: lbxxxx2@http://news1.iwon.com/odd/article/id/46354|oddlyenough|08-26-
2002::09:20|reuters.html
:2002082620020827: lbxxxx2@http://news1.iwon.com/odd/article/id/94693|oddlyenough|08-23-
2002::12:47|reuters.html
:2002082620020827: lbxxxx2@http://news1.iwon.com/index/id/oddlyenough|reuters.html
:2002082620020827: lbxxxx2@http://home.iwon.com/index.html?PG=global?SEC=bnav
:2002082620020827:
lbxxxx2@http://bfc.iwon.com/ad/in/index/id/120.html?ad_pos=INT_SLICKSTREET_PICK7&
:2002082620020827: lbxxxx2@http://lotto.iwon.com/pick7/game.jsp
:2002082620020827: lbxxxx2@:Host: lotto.iwon.com
:2002082620020827: lbxxxx2@http://lotto.iwon.com/pick7/confirm.jsp
:2002082620020827:
lbxxxx2@http://a.tribalfusion.com/p.media/ESLLSSCCJPJKGJLRHTNMTLNEORRRQMDRTXDCKL...
:2002082620020827: lbxxxx2@:Host: www.accuweather.com
:2002082620020827: lbxxxx2@http://www.aol.com/index.adp?siteId=aolcomprod&siteState=
:2002082620020827: lbxxxx2@http://specialoffers.aol.com/specialoffers/me_a.adp
:2002082620020827: lbxxxx2@:Host: specialoffers.aol.com
:2002082620020827: lbxxxx2@http://www.aol.com
:2002082620020827: lbxxxx2@:Host: www.aol.com
:2002082620020827:
bxxxx2@http://my.screenname.aol.com/_cqr/login/login.psp?siteId=atlasaol&authLev=
1&mcState=initialized&triedAimAuth=y
:2002082620020827: lbxxxx2@:Host: my.screenname.aol.com
:2002082620020827: lbxxxx2@:Host: webmail.aol.com
:2002082620020827: lbxxxx2@http://webmail.aol.com/dig/pipeCommand.adp
:2002082620020827: lbxxxx2@http://webmail.aol.com/dig/msglistframeset.adp
:2002082620020827: lbxxxx2@:Host: www.madamemercury.com
:2002082620020827: lbxxxx2@http://www.madamemercury.com/shop.html
:2002082620020827: lbxxxx2@http://www.madamemercury.com/welcome.shtml
```

I noted that the user spent a good deal of time at music/disc jockey locations, on free e-mail sites, and apparently has AOL instant messenger set up based on some of the above URLs. He also checked the weather that day. It was apparent from some of the URLs that the user had logged into the site as opposed to just visiting. There are, for example, a number of listed “messages” he went to view at the pub175.ezboard.com bulletin board site. In addition, there is a listing of webmail at AOL he apparently opened and viewed. (For sake of brevity as well as obscurity, I have eliminated a number of other entries that give more detail, as well as the ends of some of the lines above that would have given further information. In an actual investigation, this information would be preserved and carefully archived as needed. I must admit, though, I was beginning to wonder at this point whether the user accomplished much work when so much of his time was spent at these sites. There could be a case made here for a reprimand of the employee based on the amount of time he was spending at non-work related sites. This misuse of company assets would be reported to his manager for resolution. This also suggests a review and possibly update of policy and procedures in regard to internet usage. It certainly points out a need for education for corporate users to remind them of such policies.

The user was probably also unaware that he was being tracked by such ad trackers as tribal fusion and ad.trafficmp.com. There were also references to ad.doublclick.net in other index files. His browser was apparently sent to those locations at some point by whatever site he was visiting.

- Ibxxxx2\Temporary Internet Files\Content.ie5\q6yeu4nx

The content.ie5 directory almost always reveals interesting information about the user. The pictures viewed, the html files, scripts, and a host of other information can be gleaned from this location. If, for example, this user had been visiting porn sites, we would have been able to locate the pictures viewed in these directories. In this case, there was nothing so exciting as all that. Apparently the user owns or has a passion for Nissan 4 wheelers, because there were numerous pictures of those along with detailed pictures of parts, close-ups, and even a picture of the inside of the fuel door showing the sign stuck there. Looked like a lot more about Nissan four wheelers than most would ever want to know.

Finally, I reviewed files to determine the system version and service pack information that was installed on this disk. I found the $winnt.inf$, located in the WINNT\System32 directory and written 3/17/00, specified that this was not a win95, win98, or NT upgrade. That indicates the install was a complete install rather than an upgrade. A Windows Update Setup Files directory off of WINNT indicated that IE 6 had been installed as of 2/10/02 – date of the creation of that directory. And a dead directory off of root called NT4sp6ah dated 7/30/01 indicated that service pack 6 had been installed on the system as well.

In an actual investigation, the forensics experts would be taking careful notes, making printouts, and creating archives for possible litigation. Any of the above information would have been carefully documented for the courtroom and stored in a safe location for chain of evidence.

The File Type sort yielded the following information:

```
Results Summary
Images
    * /Locker//021632/021632Host/images/image_021632_hdc1

Files (64424)

    * Allocated (39860)
* Unallocated (24564)

Files Skipped (34631)

    * Non-Files (34631)
* 'ignore' category (0)

Categories (29793)

    * archive (702)
    * audio (85)
    * compress (74)
    * crypto (1)
    * data (7437)
    * disk (1)
    * documents (614)
    * exec (5188)
    * images (7278)
    * system (2567)
    * text (4976)
    * unknown (867)
    * video (3)
```

Finally, after the media analysis was complete, a new md5 sum was taken from the two partitions and compared to the original sums. They were not changed from the numbers listed above. This indicates that the evidence has not been altered, even in the test files. As the hard disk has been removed from the computer and locked away, it will not have changed, either.

Timeline Analysis

Backing out to the Host Manager, I clicked on File Activity Time Lines. First I clicked on Create Data File. This ran fls and ils. After creating a data file, I chose Create Timeline to see what file activities had taken place in chronological order. I asked for a report of activity from 1/1/2002 through 1/1/2003 because of the suspicion that the computer was no longer used after August of 2002. This choice of dates would give me the latest time the drive was used. According to the resulting file, the last use was Wednesday, August 28, 2002 at 16:58:32. At that time, the WINNT\system32\config\software and software.log files were both modified on the FAT partition. My assumption would be that was during shutdown of the computer.

A File Activity Time Line was created for the NTFS partition. A check of that partition indicated last use of 8/28/2002 at 15:58:03 when Microsoft Office was doing a search on the drive for files. In addition, an interesting phenomenon arose in that the earliest files on the NTSF partition actually dated back to 1996. This would seem to indicate that the user transferred his data off of the hard drive from an older computer to the new one, once the new one had arrived. That may explain the fact that he had both a FAT and an NTFS partition. The NTFS system was the older computer, and the FAT partition was in existence only in the new computer.

Scanning back through the most recent activities, the only actions seemed to be associated with Microsoft Office, Browsing the Internet, or in reinstalling Windows. There was nothing in the time line indicating any unusual activity. The time lines are attached to this submission in text files. The FAT timeline runs, as stated above, from 1/2002 until 8/2002. However, the NTFS timeline runs from 1996. There is actually a good deal less data on the NTFS partition. That is probably due to it not being used as much as the FAT partition after the user received his new computer.

By reviewing MACTime data, I located a list of created WINNT directories that were all created at the same time. This indicates that Windows NT was installed on March 17, 2000 at 15:53:42, the same time the directories were all created.

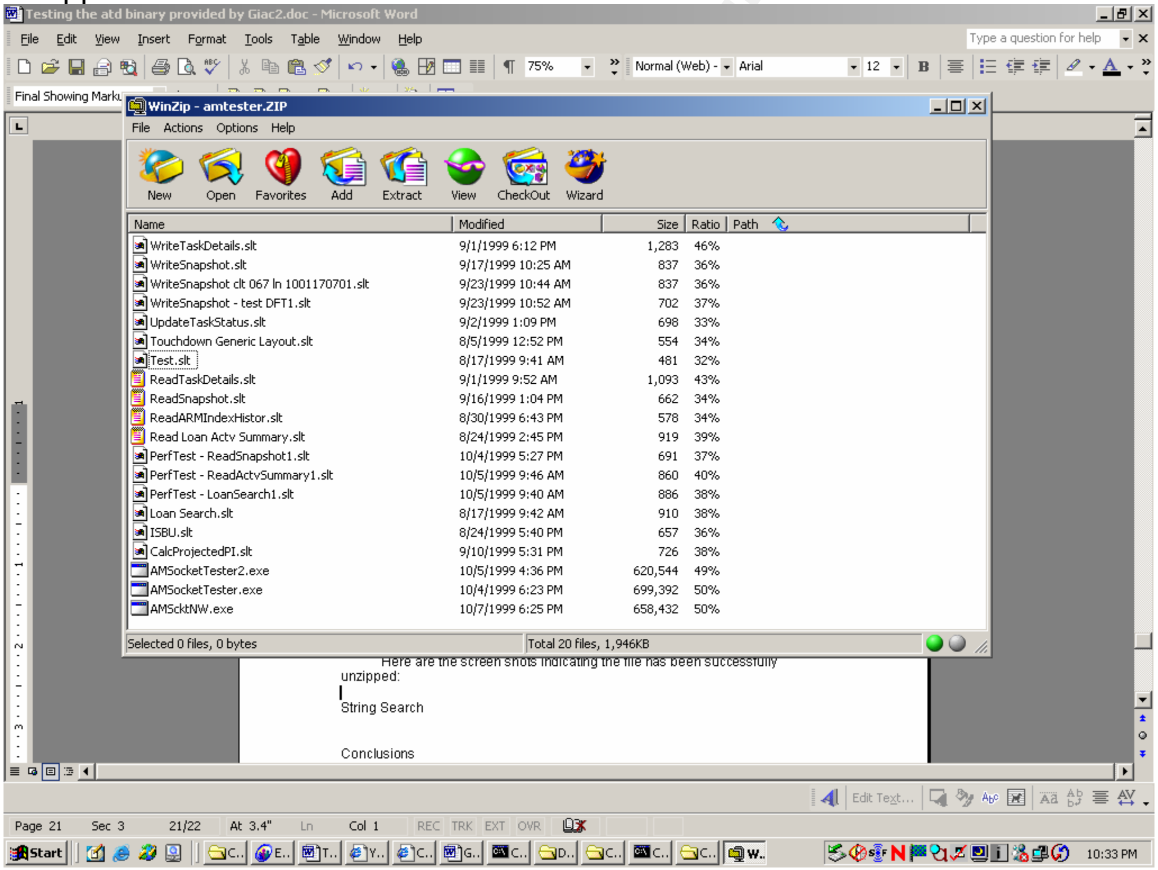
Recover Deleted Files

An attempt was made to recover several deleted files. As would be predictable, those highlighted in bright red are not recoverable. We attempted to recover several wav files and an Excel spreadsheet. However, a deleted zip file, called amtester.ZIP, created 10/15/1999, was recovered, opened with Winzip, and the files were unzipped to a local directory without error.

The files were recovered using the “Export” function in Autopsy. The zip file was selected, then I clicked on export and renamed the file to its original name. Once recovered on the Linux computer, I FTPd the file back to my Windows computer and ran Winzip to view and unzip the file. So to recover a file, the Autopsy user would:

- Click on any desired file while viewing the disk image
- Click on the Export URL
- On the Download screen, select save file to disk, then click on OK
- In the dialogue window, enter the name of the file path where you want the file to be recreated. You will probably want to name it something logical as Ethereal tends to fill that field with a long string of characters describing where the file originated. If you are recreating a file, you may just use the true name of the file, if you wish.
- Click on Save.

Here are the screen shots indicating the file has been successfully unzipped:


Testing the atd binary provided by Giac2.doc - Microsoft Word
File Edit View Insert Format Tools Table Window Help
Type a question for help
75%
Normal (Web)
Arial
12
Final Showing Markup
WinZip - amtester.ZIP
File Actions Options Help
New
Open
Favorites
Add
Extract
View
CheckOut
Wizard

Name	Modified	Size	Ratio	Path
WriteTaskDetails.slt	9/1/1999 6:12 PM	1,283	46%	
WriteSnapshot.slt	9/17/1999 10:25 AM	837	36%	
WriteSnapshot clt 067 ln 1001170701.slt	9/23/1999 10:44 AM	837	36%	
WriteSnapshot - test DFT1.slt	9/23/1999 10:52 AM	702	37%	
UpdateTaskStatus.slt	9/2/1999 1:09 PM	698	33%	
Touchdown Generic Layout.slt	8/5/1999 12:52 PM	554	34%	
Test.slt	8/17/1999 9:41 AM	481	32%	
ReadTaskDetails.slt	9/1/1999 9:52 AM	1,093	43%	
ReadSnapshot.slt	9/16/1999 1:04 PM	662	34%	
ReadARMIndexHistor.slt	8/30/1999 6:43 PM	578	34%	
Read Loan Actv Summary.slt	8/24/1999 2:45 PM	919	39%	
PerfTest - ReadSnapshot1.slt	10/4/1999 5:27 PM	691	37%	
PerfTest - ReadActvSummary1.slt	10/5/1999 9:46 AM	860	40%	
PerfTest - LoanSearch1.slt	10/5/1999 9:40 AM	886	38%	
Loan Search.slt	8/17/1999 9:42 AM	910	38%	
ISBU.slt	8/24/1999 5:40 PM	657	36%	
CalcProjectedPI.slt	9/10/1999 5:31 PM	726	38%	
AMSocketTester2.exe	10/5/1999 4:36 PM	620,544	49%	
AMSocketTester.exe	10/4/1999 6:23 PM	699,392	50%	
AMScktNW.exe	10/7/1999 6:25 PM	658,432	50%	

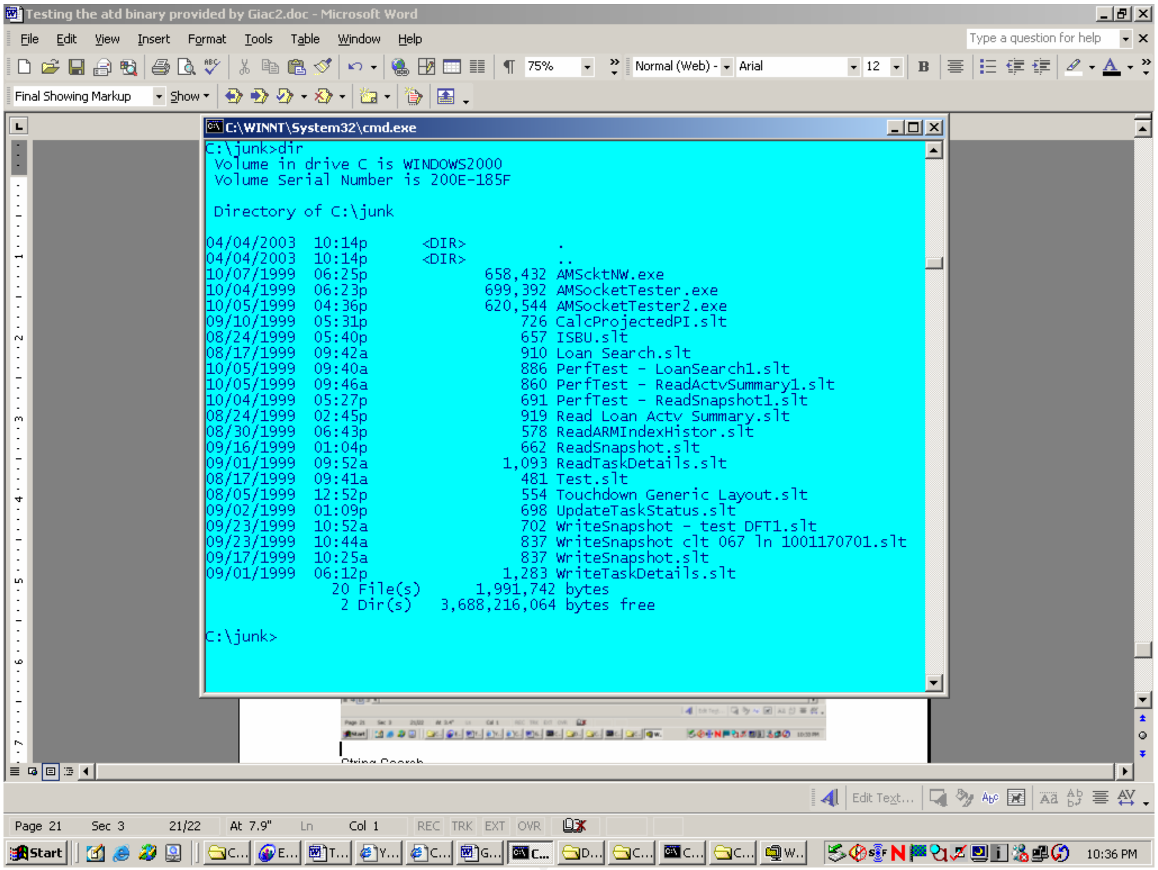
Selected 0 files, 0 bytes

Total 20 files, 1,946KB

Here are the screen shots indicating the file has been successfully unzipped:

String Search

Conclusions


Testing the atd binary provided by Giac2.doc - Microsoft Word
File Edit View Insert Format Tools Table Window Help
Type a question for help
75%
Normal (Web)
Arial
12
Final Showing Markup
Show
C:\WINNT\System32\cmd.exe
Edit Text...
Page 21 Sec 3 21/22 At 7.9" Ln Col 1 REC TRK EXT OVR
Start
10:36 PM

```
C:\junk>dir
 Volume in drive C is WINDOWS2000
 Volume Serial Number is 200E-185F

 Directory of C:\junk

04/04/2003  10:14p      <DIR>          .
04/04/2003  10:14p      <DIR>          ..
10/07/1999  06:25p             658,432 AMScktNW.exe
10/04/1999  06:23p             699,392 AMSocketTester.exe
10/05/1999  04:36p             620,544 AMSocketTester2.exe
09/10/1999  05:31p                 726 CalcProjectedPI.slt
08/24/1999  05:40p                 657 ISBU.slt
08/17/1999  09:42a                 910 Loan Search.slt
10/05/1999  09:40a                 886 PerfTest - LoanSearch1.slt
10/05/1999  09:46a                 860 PerfTest - ReadActvSummary1.slt
10/04/1999  05:27p                 691 PerfTest - ReadSnapshot1.slt
08/24/1999  02:45p                 919 Read Loan Actv Summary.slt
08/30/1999  06:43p                 578 ReadARMIndexHistor.slt
09/16/1999  01:04p                 662 ReadSnapshot.slt
09/01/1999  09:52a               1,093 ReadTaskDetails.slt
08/17/1999  09:41a                 481 Test.slt
08/05/1999  12:52p                 554 Touchdown Generic Layout.slt
09/02/1999  01:09p                 698 UpdateTaskStatus.slt
09/23/1999  10:52a                 702 WriteSnapshot - test DFT1.slt
09/23/1999  10:44a                 837 WriteSnapshot clt 067 ln 1001170701.slt
09/17/1999  10:25a                 837 WriteSnapshot.slt
09/01/1999  06:12p               1,283 WriteTaskDetails.slt
              20 File(s)      1,991,742 bytes
               2 Dir(s)   3,688,216,064 bytes free

C:\junk>
```

String Search

A word search was conducted in Autopsy for expressions that might be associated with malware indicated that the user had neither purposely installed such software, nor had his computer been compromised by such. Some of the word searches included common "dirty" words – required vocabulary of many hackers – and words, file names, and phrases associated with malware. For example, a search for:
PRIVMSG, %s, ctcp, <nick>, 848348, getnick, getnonick, rnick , sacker, addy, jacker, stopsack, stopjack, spawn, randnick, clone, clonedie
All known to be associated with a Trojan horse, (srvcp.exe), per Lenny Zeltser in his article "Reverse Engineering Malware" located at:
http://www.megasecurity.org/Info/Reverse%20Engineering%20Malware.htm

The following text files were chosen for a visual search because they have significance to the system or because they appeared to have potential importance. They were searched for inappropriate or unusual entries and none were found indicating that these system files had not been tampered with:

- Win.ini (system file for Windows)
- System.ini (system file for Windows)
- Setihome.ini (looks like he was into the Seti project.)
- Schedlog.txt (ran regularly, but executed no software per the log.)
- Runonceex_log.txt (looked like it might contain log information)
- Autoexec.bat (system file)

- Config.sys
- Wininit.ini
- Autostart folder

The registry could not be examined with the tools used/provided by SANS, as we were operating on a Linux computer and the drive was NTFS. To do a more thorough test, the registry would need to be viewed/dumped and examined using a Windows system and appropriate tools. As it was expected that this computer hard drive would not yield any sign of malfeasance, we did not/have not yet acquired the tools we need to view the registry information.

Conclusions

Our user has not been a total model citizen, but he's been pretty good. The Antivirus software ran regularly on his computer, as indicated by Norton Antivirus temp files being deleted on a regular basis and remaining in the directory for us to find in our file search. In addition, all of the virus software had been accessed the last day of use, indicating that the checker was turned on.

The user is a developer who uses java developer kits and Delphi tools to create software for the corporation. These are tools that are approved and appropriately licensed. He also pointed to a couple of MainFrames on a regular basis, as he had added their names and IP addresses in the host table. This was reasonable based on his job function.

Our user did surf the net regularly as indicated by many files listed above showing places he had been and images he has viewed. While some of his browsing may not have been appropriate during work time, there may not be enough of an offense here for more than a reprimand. Certainly not enough to call for termination.

There is no sign of any malware or serious misuse of company assets on the investigated drive.

The most serious problem found on this computer was the proprietary information that should not have been there were it to be given away or discarded. This information encourages the enforcement of policies governing old computers. LAN/Desktop should systematically "wipe" all data off of every computer taken out of service as it is removed. This way a computer won't sit around so long that someone will forget what kind of valuable data resides on it.

Part 3: Legal Issues of Incident Handling

I run an Internet Service Provider (ISP) and have paying customers. A (verified) law enforcement officer has called me to inform me that an account on my system was used to hack into a government computer. He has asked that I verify the activity by reviewing my logs and determine if my logs reflect whether or not the activity was initiated from my server(s), or from another upstream provider. I have reviewed the logs, but can only determine that a valid user logged in via a dialup account during the period of the suspicious activity.

Questions:

A. What, if any, information can I provide to the law enforcement officer over the phone during the initial contact?

The answer to this question will rely to a great degree on the current policies of my corporation. In a case where I run an ISP, the policy may be more permissive while the policy at a financial institution or other business that would stand to lose reputation based on such a breach of their system may choose to not share such information unless forced by a subpoena.

Corporate policies are, (or should be), based on legal precedence. Health related companies have the additional guidance of HIPAA, for example. All companies have to answer to acts that protect a consumer's privacy. The Graham, Leach, Bliley law, affectionately known as GLB, governs all financial institutions as well as their service providers. Our corporation may be providing service(s) to a financial institution or a heath institution and is, therefore, very protective of end consumer's information due to the possibility of litigation.

In addition to the complication of law, there is also reputation and future business on the line. A corporation can lose many customers and much potential business even because of what is a perceived security issue. Reputation is crucial to any business.

A careful balancing act is performed by most businesses between what is required by law, due diligence, what is best for business, what is best for our customer, and what is best for the end consumer who may or may not be our actual customer. This balancing act, as much as is practical, is documented in every well run business' policies and procedures. This is to minimize the exposure to the business by well meant but poorly executed acts – such as inappropriate personnel talking to media and/or to public officials at the wrong time or giving out damaging information – damaging either to our business, our client, and/or the end consumer. A thoughtless, undocumented process will send the best of businesses straight to the toilet.

My answer to the officer on the initial call will probably be to assure him that allowing such activities is definitely against our Information Security Policies and that we will cooperate with the appropriate law enforcement agencies as required by law. I would let him know how long I thought it would take to review the logs and determine the answer(s) to his questions, and that I would be in touch with him within that time frame.

If the officer can provide information indicating the ID or other information on the hacker, it may be simple to find out if those users are currently on line or when was the last time they accessed the system. Some of what we can tell them will depend upon what they have discovered and can tell us. If he can give us the known times of the attacks, this information will be invaluable in determining whose account may have been used for the attacks, what software was used in the attack, etc.

Finally, what I can tell him during any call is going to depend greatly on how good a job we are doing reviewing the logs. One would hope that, with IDS, firewalls, etc., that there would be every opportunity to discover this issue before being called by the law enforcement agency. In that case, it would behoove the ISP to make first contact and share what we have learned with law enforcement.

B. What must the law enforcement officer do to ensure that I preserve this evidence if there is a delay in obtaining any required legal authority?

The officer must let me know as soon as possible what information he will require. Then, all logs and associated records that may be helpful can be archived for an indefinite period of time and set aside for future research. Time is of the essence as company policy will dictate a data retention period that may be shorter than the requirements of litigation. That is, we will normally save data a reasonable period – long enough to adequate analysis in case of a suspected breach of security. We will not typically save log data forever because there is no return on the investment required for such data storage. Our ISP may save logs for as much as 90 days or even 6 months, but litigation could take years. The sooner we can be notified that certain logs need to be archived indefinitely, the more likely we are to be able to provide that service.

Law requires the government to request we retain data for a period of 90 days per the following information from http://www.usdoj.gov/criminal/cybercrime/ECPA2701_2712.htm:

> **(f) Requirement to preserve evidence.**--
>
> **(1) In general.**--A provider of wire or electronic communication services or a remote computing service, upon the request of a governmental entity, shall take all necessary steps to preserve records and other evidence in its possession pending the issuance of a court order or other process.
>
> **(2) Period of retention.**--Records referred to in paragraph (1) shall be retained for a period of 90 days, which shall be extended for an additional 90- day period upon a renewed request by the governmental entity.

Again, depending upon the policies of the company, we will probably require that the officer present us with a formalized, legal request for such data, such as a court order, before releasing any such logs. However, in the sense of cooperation with the authorities, we could archive the data requested without releasing it before the court order is delivered. (See C.)

To adequately address the officer's questions, we may need to make complete hard drive images or provide other methods of gathering evidence that can be used for forensics purposes. As business will go on, one would assume that the ISP would be responsible for protecting the chain of evidence while rebuilding his systems for use on the network.

C. What legal authority, if any, does the law enforcement officer need to provide me in order for me to be able to send him my logs?

Typically, a law enforcement person will obtain a court order or a search warrant to obtain such information. Companies do not, typically, allow their logs to be disseminated in public. However, if such logs become evidence in conjunction with some legal action, all the rules and statutes pertaining to evidence will govern the acquisition of such evidence by public officials. Due diligence, then, requires that a business exercise care in collecting and maintaining such log material. Until the public officials acquire the evidence, it should be understood by our ISP business that we could be considered accomplices to the act, should we not protect the logs in a reasonable manner. That is, if the logs become corrupt, polluted, or cannot be substantiated as not having been tampered with, it would be reasonable for the court to hold the ISP liable as an accessory to the crime itself by suppressing evidence. If we do not care about the effect of the crime upon the victim, we become suspect.

At http://www.cybercrime.gov/PatriotAct.htm, one will find the following information concerning electronic evidence under this section title:

Section 210 Scope of Subpoenas for Electronic Evidence

…

Amendment: Amendments to section 2703(c) update and expand the narrow list of records that law enforcement authorities may obtain with a subpoena. The new subsection 2703(c)(2) includes "records of session times and durations," as well as "any temporarily assigned network address." In the Internet context, such records include the Internet Protocol (IP) address assigned by the provider to the customer or subscriber for a particular session, as well as the remote IP address from which a customer connects to the provider. Obtaining such records will make the process of identifying computer criminals and tracing their Internet communications faster and easier.

Moreover, the amendments clarify that investigators may use a subpoena to obtain the "means and source of payment" that a customer uses to pay for his or her account with a communications provider, "including any credit card or bank account number." 18 U.S.C. §2703(c)(2)(F). While generally helpful, this information will prove particularly valuable in identifying the users of Internet services where a company does not verify its users' biographical information. (This section is not subject to the sunset provision in section 224 of the Act).

The above information clearly identifies log type data (IP addresses, session times and durations), and specifies that it can be subpoenaed. Again, it is the ISP's responsibility to maintain such data for a reasonable period of time as outlined by their policies, and to archive such data for a longer period if we suspect it will be used in an investigation.

D. What other "investigative" activity am I permitted to conduct at this time?

In the mean time, reasonable monitoring should be continued by the ISP to protect the provider's rights and property and/or as a necessary part of providing such services to users in order to ensure quality of service.

E. How would my actions change if my logs disclosed a hacker gained unauthorized access to your system at some point, creating an account for him/her to use, and then used THAT account to hack into the government system?

I am assuming in this scenario that the agency whose system was breached would already know since the law enforcement person contacting me was aware of the issue. I do not believe my actions should be any different whether the breach affected government or private citizens. We should still do due diligence with our evidence, and we should still cooperate within the parameters of the law.

References Cited

Computer Crime and Intellectual Property Section (CCIPS). “Field Guidance on New Authorities That Relate to Computer Crime and Electronic Evidence Enacted in the USA Patriot Act of 2001.” 5 November 2001. URL: http://www.cybercrime.gov/PatriotAct.htm (3 April 2003).

Dashie. “BFi numero 7, anno 2 - 25/12/1999 - file 13 di 22.” 25 December 1999. URL: http://www.s0ftpj.org/bfi/online/bfi7/bfi07-13.html. (23 May 2003).

Department of Justice. “UNITED STATES CODE ANNOTATED TITLE 18. CRIMES AND CRIMINAL PROCEDURE PART I—CRIMES CHAPTER 121--STORED WIRE AND ELECTRONIC COMMUNICATIONS AND TRANSACTIONAL RECORDS ACCESS “ 19 May 2003. URL: http://www.usdoj.gov/criminal/cybercrime/ECPA2701_2712.htm. (24 May 2003).

Farmer, Dan. “Lazarus.” 19 August 2000. URL: http://staff.washington.edu/dittrich/talks/blackhat/tct/man/man1/lazarus.1.html (4 April 2003).

Garner, Jr., George M. “Forensic Acquisition Utilities.” 25 April 2002. URL: http://users.erols.com/gmgarner/forensics/. (23 May 2003).

Jerico, null, and Munge. “Attrition Web Page Hack Mirror.” No publish date indicated. URL: http://www.attrition.org/mirror/ (23 May 2003).

Low, Christopher. “ICMP Attacks Illustrated.” SANS Infosec Reading Room. 11 December 2001. URL: http://www.sans.org/rr/threats/ICMP_attacks.php (3 April 2003).

Staff writers. “Packet Storm.” May 23 11:27:39 2003. URL: http://packetstorm.linuxsecurity.com/ (23 May 2003).

Staff writers. “Phrack”. 28 December 2002. URL: http://www.phrack.org/. (23 May 2003)

Staff writers. “SourceForge.” 23 May 2003. URL: http://sourceforge.net/. (23 May 2003).

Veillard , Daniel.   “task-1.52-2 mdk RPM for i586.” 6 January 2003. URL: http://rpmfind.net/linux/RPM/cooker/contrib/i586/task-1.52-2mdk.i586.html (4 April 2003)

Zeltser, Lenny. “Reverse Engineering Malware.”   May 2001. URL: http://www.megasecurity.org/Info/Reverse%20Engineering%20Malware.htm (4 April 2003)