GIAC Certified Forensic Analyst
Practical Assignment Version 1.4

Resubmitted By
Jennie Callahan
23 January, 2004

ABSTRACT

This report was prepared to satisfy the requirements of the GIAC Certified Forensic Analyst practical exercise. The report is divided into three parts:

Part 1: Describes the analysis of the SANS provided unknown binary file to include the steps taken to determine the contents/process of the file and demonstrating analyst's ability to determine what the unknown file is capable of performing.

Part 2: A forensic examination of an unknown computer. The forensic examination includes a description of the case, the computer, and the evidence found on a compromised computer. It also includes a timeline analysis, recovery of deleted files, and a string search. Each of the elements covered was designed to establish the examiners knowledge of forensic protocols and examinations.

Part 3: Discusses legal issues of computer forensic investigation. Several questions were presented in which the answers required research of the laws relevant to the examiner.

Each of the portions were designed to test the amount of knowledge retained from the course as well as procedural issues which might be of assistance.

Part I – Analyze an Unknown Binary

Binary Details:

1. The true name of the unknown binary “prog” was bmap-1.0.20. The true name was determined by using the Linux strings command to find text unique to the program. In a Linux command prompt, the command “strings” was executed to determine specific characteristics and keywords for the program. Once keywords were identified inside of the program, an Internet search was conducted in an effort to identify similar keywords associated with other programs. The keywords used were noted in question number 6. The Internet search revealed several hits describing the use of a file called bmap. Attributes specific to the bmap program were described as a blocksize of a typical file system varies from 1K to 4K. Every file takes at least one block. The unused space in that block is slack space. Bmap can save data into this slack space, extract data from slack space, and delete data in slack space. The data cannot be accessed using tools unaware of slack space (ie. almost all other tools). The prog and bmap files, does not change existing files, and therefore cannot be detected using checksums or access times. Additional internet research into the file bmap as well as a strings search and program testing of the bmap file and the prog file revealed the two files were similar and used for the same purpose. See the Program Description, Forensic Details and Program Identification Sections for the full processes used in identification of the prog file as bmap.

2. The prog file’s last modified time was Mon Jul 14 14:24:00 2003. The prog file’s last accessed time was Wed Jul 16 06:12:45 2003 and it was last changed on Wed Jul 16 06:05:33 2003. The access time of a file showed the approximate time when the file was last accessed. The modified time of a file showed the time the data was last modified. The changed time of a file showed the time the inode record was last modified. The prog file’s modified, accessed and changed times were obtained using the forensic utility Sleuth Kit and Autopsy v. 1.70.

The Sleuth program, known as Sleuth Kit (previously known as TASK) is a collection of UNIX-based command line file system and media management forensic analysis tools. The file system tools allow you to examine NTFS, FAT, HFS, EXT2, and EXT3FS file systems of a suspect computer in a non-intrusive fashion. The tools have a layer-based design and can extract data from the internal file system structures. Because the tools do not rely on the operating system to process the file systems, deleted and hidden content is shown. The media management tools allow you to examine the layout of disks and other media. With these tools the locations of partitions are located and extracted so that they can be analyzed with file system analysis tools.

Performing a complete analysis of a system using command line tools can become tedious. Supporting Sleuth is the Autopsy Forensic Browser which is a graphical interface to the tools in The Sleuth Kit, and allows the easy conduct of an investigation. Autopsy provides case management, image integrity, keyword searching, and other automated operations and both the Sleuth Kit and Autopsy are open source and free to download. Their combined features include: View Allocated and Deleted Files and Directories; Access to low-level file system structures; Timeline of file activity; File category sorting and extension checking; Keyword searches including grep regular expressions; Graphic image identification and thumbnail creation; Hash database lookups including the NIST NSRL and Hash Keeper; Investigator notes and Report generation. The Sleuth Kit is written in C and Perl and uses some code and design from The Coroner's Toolkit (TCT).

The dd image of the floppy disk provided was mounted into Sleuth Kit for analysis. Autopsy provided a graphical interface (web browser) to review the results of the Sleuth Kit tools. Specifically within the Sleuth Kit, the tool “mactime” was used to create a timeline of the file activity which was generated from the kit command fls. The command fls read and displayed the file directory entries in a directory inode. Figure 1.1 was a screen shot showing the Autopsy interface with a mounted image and also included the MAC times of the prog file highlighted in blue.

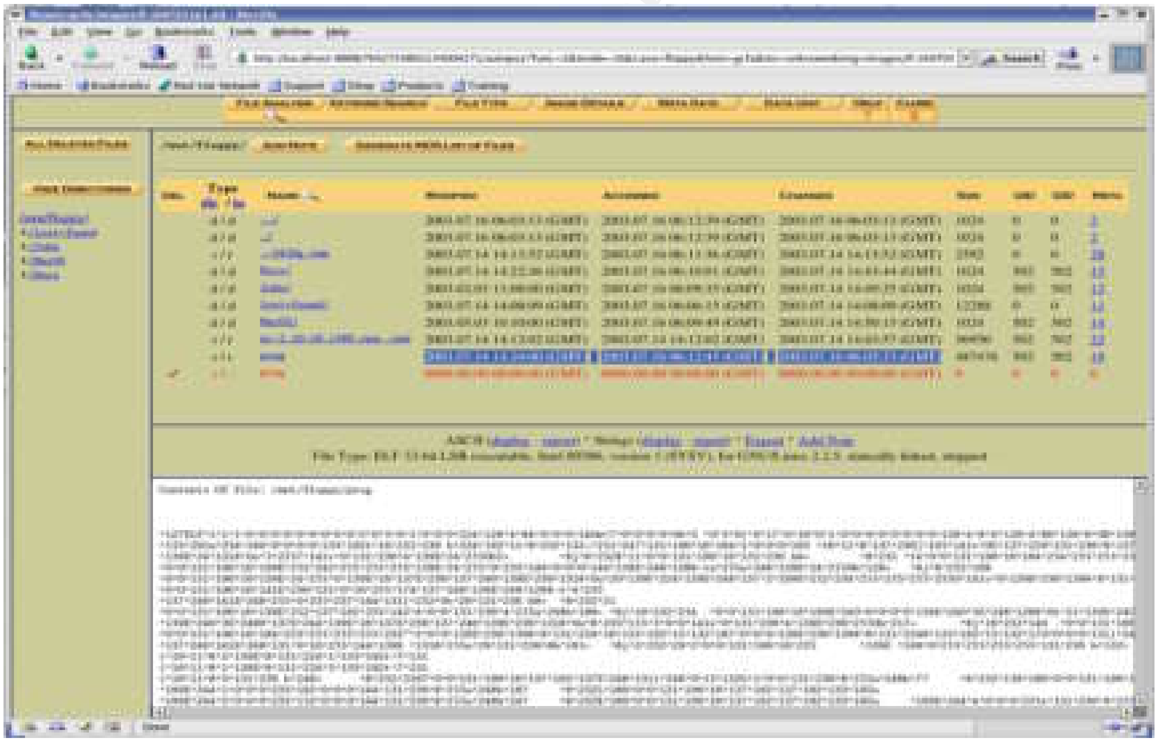


Figure 1.1

3. The program file’s Owner and Group Identification Numbers were 502 and 502. The owner of the file is stored in the inode table in the form of a number. The number corresponds to the User ID (UID) and Group ID (GID) number from host system’s /etc/passwd file. The UID and GID numbers 502 of the prog file were obtained from the inode table on the floppy disk and noted in the Autopsy

interface. See the UID and GID columns from Figure 1.1 above.

4. The program file size was 487476 bytes. The inode table also contains information on what blocks of a device are occupied by a particular file called pointers. Linux uses a data block (sector/cluster) as its basic unit and can the block size can be 512-4096 bytes. The physical file size, the amount of space used on the disk, is determined by multiplying the number of blocks used by the block size. In the case of the prog file 77 blocks were used, and the block size of the floppy disk was 1024, equaling 488448 bytes. The logical file size, the actual size of the file, was 487476 bytes. Figure 1.2 showed the pointers and file size obtained from the inode table using Autopsy. The Inode Modified Time is the same as file changed time noted earlier.

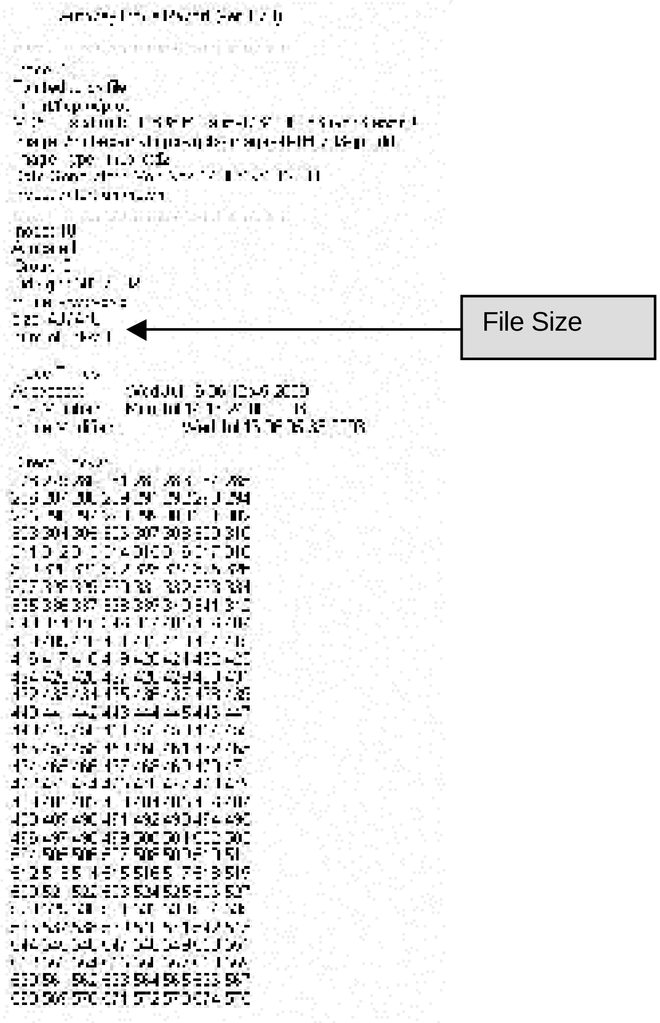
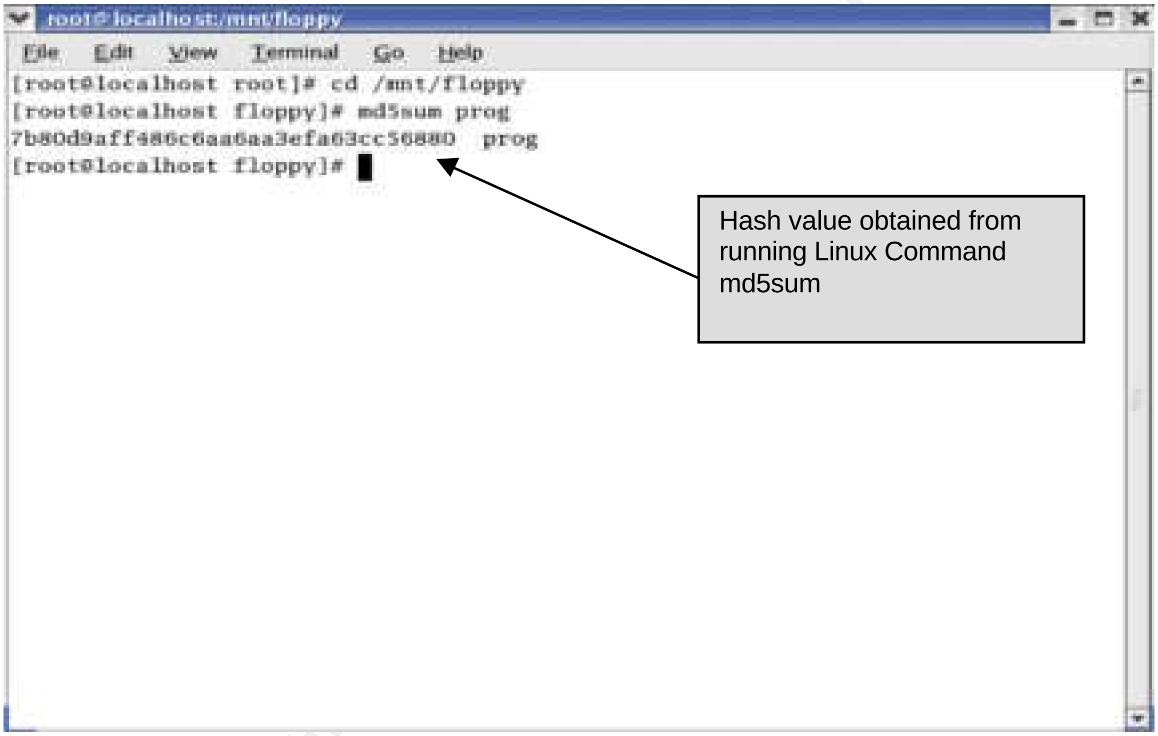


Figure1.2

5. The prog file's MD5 Hash was 7b80d9aff486c6aa6aa3efa63cc56880. An MD5 hash value is a 128-bit (16 byte) number that uniquely describes the contents of a file. It is essentially a digital fingerprint of a file or an entire disk.

For this reason, the MD5 is a standard in the forensic world. The alogorithm used to generate the MD5 hash is such that the odds of two different files having the same hash value is two to the 128th power. The MD5 hash value was developed by RSA and is publicly available.

The dd image of the floppy provided was mounted in read-only mode to the /mnt/floppy directory of a Linux forensic workstation so as the files could be viewed as a typical user without modifications to the files themselves. The MD5 Hash value of the prog file was obtained by using the Linux command md5sum against the prog file. Figure 1.3 was a screen shot of the commands used and the resulting hash value.



```
[root@localhost root]# cd /mnt/floppy
[root@localhost floppy]# md5sum prog
7b80d9aff486c6aa6aa3efa63cc56880  prog
[root@localhost floppy]# 
```

Hash value obtained from running Linux Command md5sum

Figure 1.3

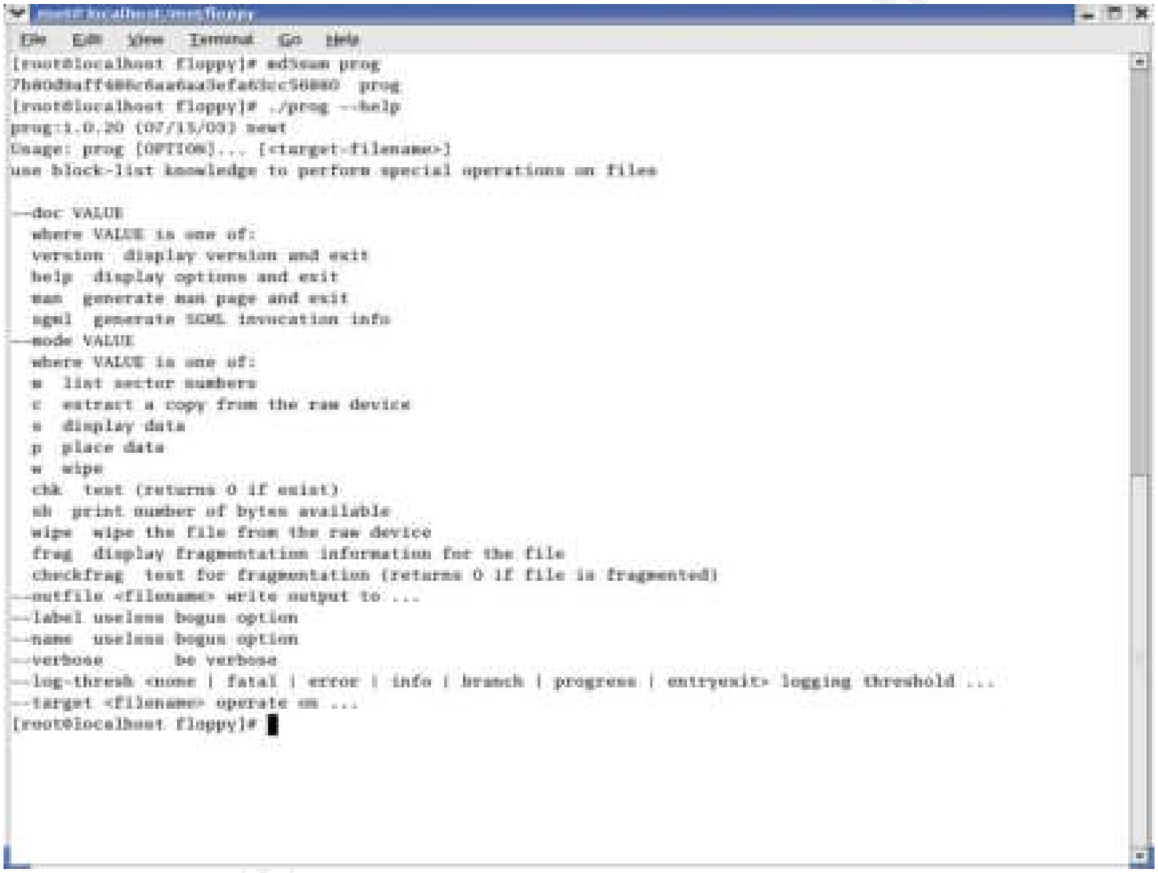
6. The keywords discovered from the prog file included “prog”, “1.0.20”, “newt”, and “use block-list knowledge to perform special operations on files”. The listed keywords associated with the prog file were discovered by using the Linux strings command against the prog file. The results of the strings command reveals the text located inside of a file. The above text was found and was noted as keywords as the text appeared to be unique to the prog file.

Program Description:

File systems use block sizes between 1K to 4K. Every file takes at least one block and the unused space in that block is slack space. The prog file was designed to save, extract and delete data in slack space. Based on the MAC times covered in the previous section, the prog file was last accessed on 16 July

2003, at 06:12:45(GMT).

The dd image of the floppy provided was mounted in read-only mode to the /mnt/floppy directory of a Linux forensic workstation so as the files could be viewed as a typical user without modifications to the files themselves. The program's help menu information was accessed by typing in the Linux command prog –help, which revealed the usage of the prog file and the various switches a user can add in an effort to run the program in varying ways. Figure 1.4 was a screen shot of the commands used and the resulting help menu displayed.

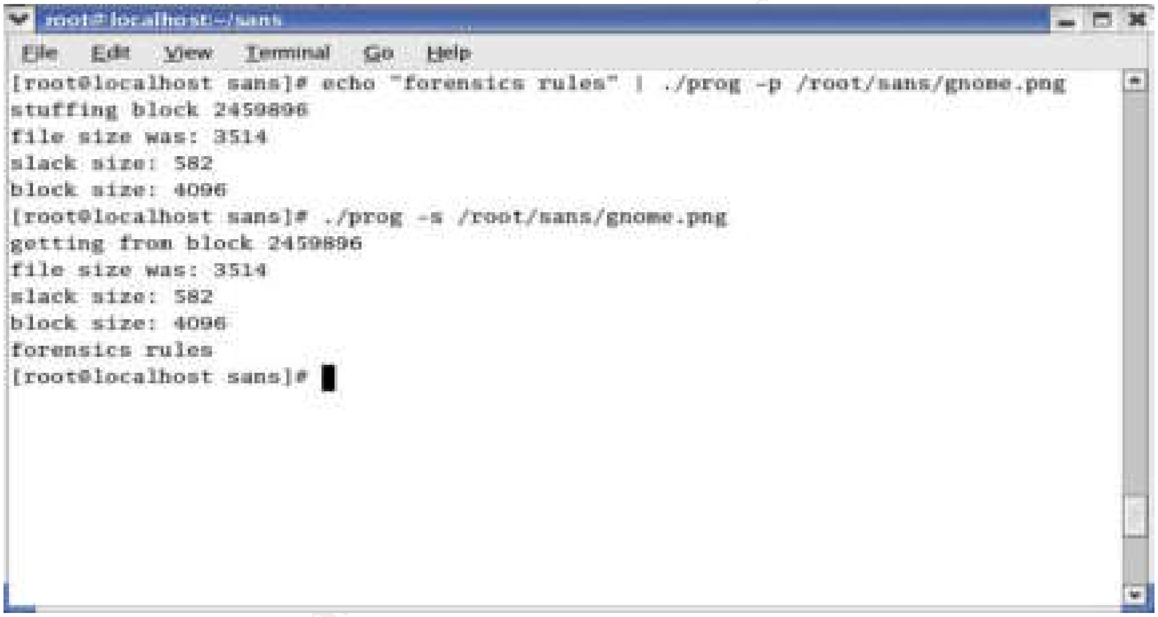


```
[root@localhost floppy]# md5sum prog
7b80d8aff486c6aa6aa3efa63cc56880  prog
[root@localhost floppy]# ./prog --help
prog:1.0.20 (07/15/03) newt
Usage: prog [OPTION]... [<target-filename>]
use block-list knowledge to perform special operations on files

--doc VALUE
  where VALUE is one of:
  version  display version and exit
  help  display options and exit
  man  generate man page and exit
  sgml  generate SGML invocation info
--mode VALUE
  where VALUE is one of:
  m  list sector numbers
  c  extract a copy from the raw device
  s  display data
  p  place data
  w  wipe
  chk  test (returns 0 if exist)
  sb  print number of bytes available
  wipe  wipe the file from the raw device
  frag  display fragmentation information for the file
  checkfrag  test for fragmentation (returns 0 if file is fragmented)
--outfile <filename> write output to ...
--label useless bogus option
--name  useless bogus option
--verbose       be verbose
--log-thresh <none | fatal | error | info | branch | progress | entryexit> logging threshold ...
--target <filename> operate on ...
[root@localhost floppy]# █
```

Figure 1.4

The help menu from the prog file showed the statements "use block-list knowledge to perform special operations on files" and "prog [OPTION]... [<target-filename>]". The two key statements were entered into a Google based internet search, which revealed web site links to pages describing the use of a program called bmap. Additional research at the listed pages revealed the program titled bmap-1.0.20 was available for download and could be used in the same method as prog. See Additional Information Section below for websites used. The bmap-1.0.20 source code and Linux RPM files were downloaded, compiled, and executed on the RedHat 9.0 Linux forensic workstation. The bmap program contained similar help information and was also utilized in the same manner as prog. When the prog program was executed, the program determined the block size, determined the given file's logical size, calculated the subsequent amount of slack space and placed the requested data into the slack space of the file. Figure 1.5 shows the commands and results of using the prog file. The first command, using the –p switch, places the text "forensic rules" in the slack space of the file entitled gnome.png. Even though the operating system believes the space is unavailable, the prog file writes the text to the actual available space in remaining portion of the block on the physical media. The prog file also discovered and displayed the logical file size of the gnome.png file as weII as the amount of slack space available for storage of the text "forensics rules". The second command, using the –s switch, revealed the text placed into the gnome.png file by the user. The program is also capable of wiping the slack area (-w switch) and checking and displaying file fragmentation information (-frag switch). The full options can be viewed in the help menu, which can be viewed in Figure 1.4.



```
[root@localhost sans]# echo "forensics rules" | ./prog -p /root/sans/gnome.png
stuffing block 2459896
file size was: 3514
slack size: 582
block size: 4096
[root@localhost sans]# ./prog -s /root/sans/gnome.png
getting from block 2459896
file size was: 3514
slack size: 582
block size: 4096
forensics rules
[root@localhost sans]#
```

Figure 1.5

Forensic Details:

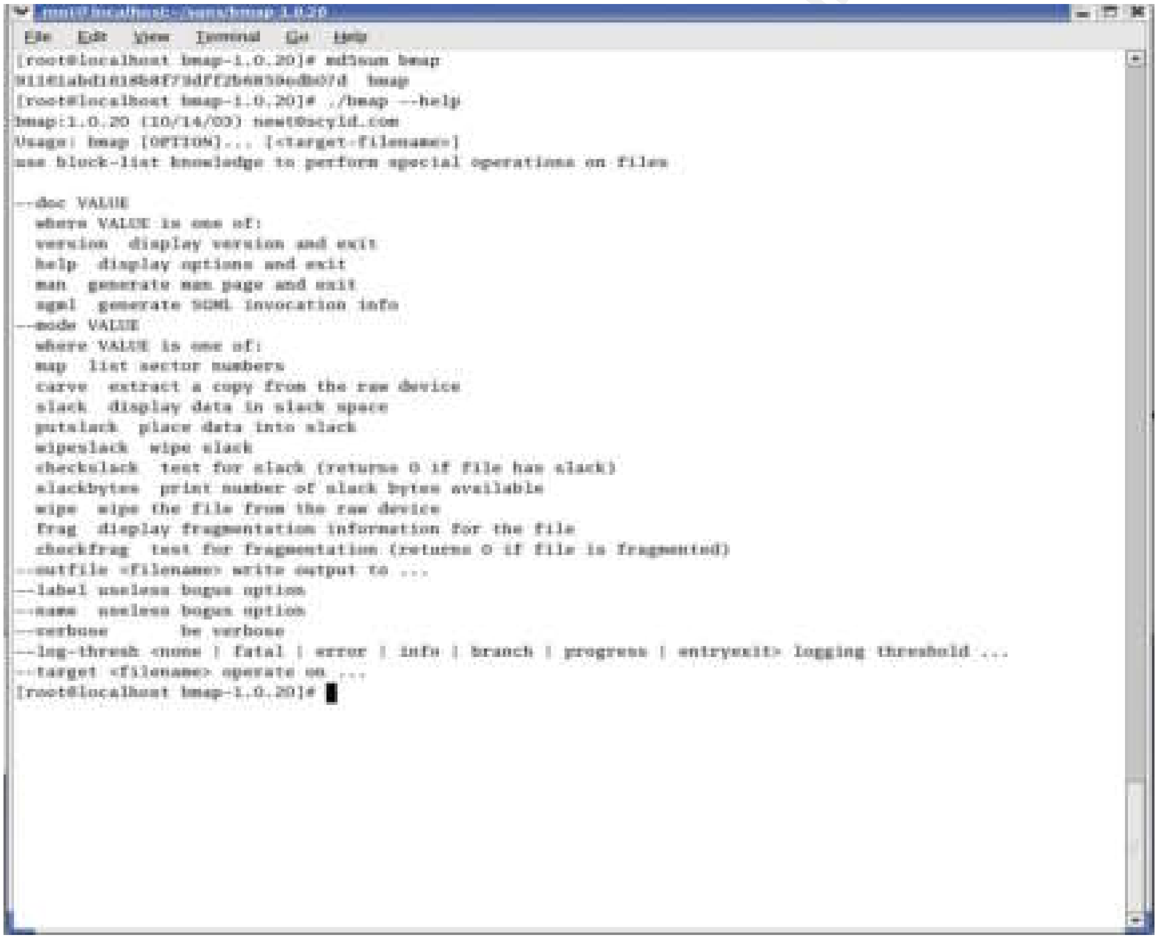
The dd image provided was mounted as read-only to the /mnt/floppy directory of a computer forensic workstation using RedHat Linux 9.0. The dd image was also mounted into the forensic utility Sleuth Kit and the Autopsy program was used to display the results of the Sleuth Kit utilities.

When the prog and bmap files were executed it was established the use of a file's slack space would not change the file itself. In the example above, the gnome.png file was not altered in any way by the use prog file. Because the alteration is to the slack space and not the file itself, the use of the programs prog and bmap cannot be detected using checksums or MAC times of the storage file. The data stored in slack space cannot be accessed using tools unaware of slack space; however, forensic tools such as Autopsy, EnCase, or bmap can read into slack space. Using the Linux "lsof" command to view processes with corresponding UID, GID, and PID, no other system files or processes were noted as being utilized by the prog or bmap programs. The Linux netstat command was used (the netstat command symbolically displays the contents of various network-related data structures). [There are a number of output formats, depending on the options for the information presented. The first form of the command displays a list of active sockets for each protocol. The second form presents the contents of one of the other network data structures according to the option selected. Using the third form, with a wait interval specified, netstat will continuously display the information regarding packet traffic on the configured network interfaces. The fourth form displays statistics about the named protocol) Ethereal Network (free network protocol analyzer for Unix and Windows which allows the examination of data from a live network or from a capture file on disk. Ethereal has several powerful features, including a rich display filter language and the ability to view the reconstructed stream of a TCP session) monitor to view network connections]. No network connections were attempted or established when the prog and bmap files were executed. The program bmap is freely available for download from the internet and was documented as being written by Daniel Ridge, newt@scyld.com. No additional investigative leads were obtained from the program itself.

Program Identification

The bmap-1.0.20 program was identified as a similar program to "prog"when the Internet research was conducted. The Internet research revealed the source code was available for download and was obtained by file transfer protocol (ftp) at ftp://ftp.scyld.com/pub/forensic_computing/bmap, which appeared to be the primary source to download the bmap files (previous versions were also available), although not the only source. The bmap file was compiled using RedHat Linux 9.0 on a forensic workstation. The hash values of the prog file and the compiled program were not the same as the systems used to compile the program were not the same. PRICE likely used another version of Linux to compile the program. If PRICE compiled the prog file found on the floppy disk from source code, the compiled program may exist on a hard drive used by him to compile the prog file. A search of his home or office computers may reveal the originally compiled program. The md5sums of the two files (the originally compiled file and the file on floppy disk) should be identical and the date stamps embedded within the help information should also match.

Similar steps were taken as previously noted in the Program Description Section to fully identify the abilities of the bmap program and compare the results to the prog file. The bmap-1.0.20 program was executed from the same RedHat 9.0 Linux forensic workstation that it was compiled on. Figure 1.6 shows the hash value obtained by using the Linux md5sums command against the compiled bmap program file. The figure also shows the contents of the bmap help menu when –help switch was used. The help menu also contained the phrase "use block-list knowledge to perform special operations on files" and "bmap [OPTION]... [<target-filename>]" which were key phrases previously noted in the prog file's help menu. The specific switches used by the bmap program were different than those listed by the prog file; however the functions of the switches were the same. In bmap, the switch to place data into slack space was –putslack and for the prog file was –p. A complete comparison of the bmap's switches compared to the switches of prog revealed the prog switches were shortened versions of the same switch and completed the same tasks as the bmap program and corresponding switches. An example of this was that the prog switch –s was the same as the –slack switch in bmap as both were used to display data in slack space.

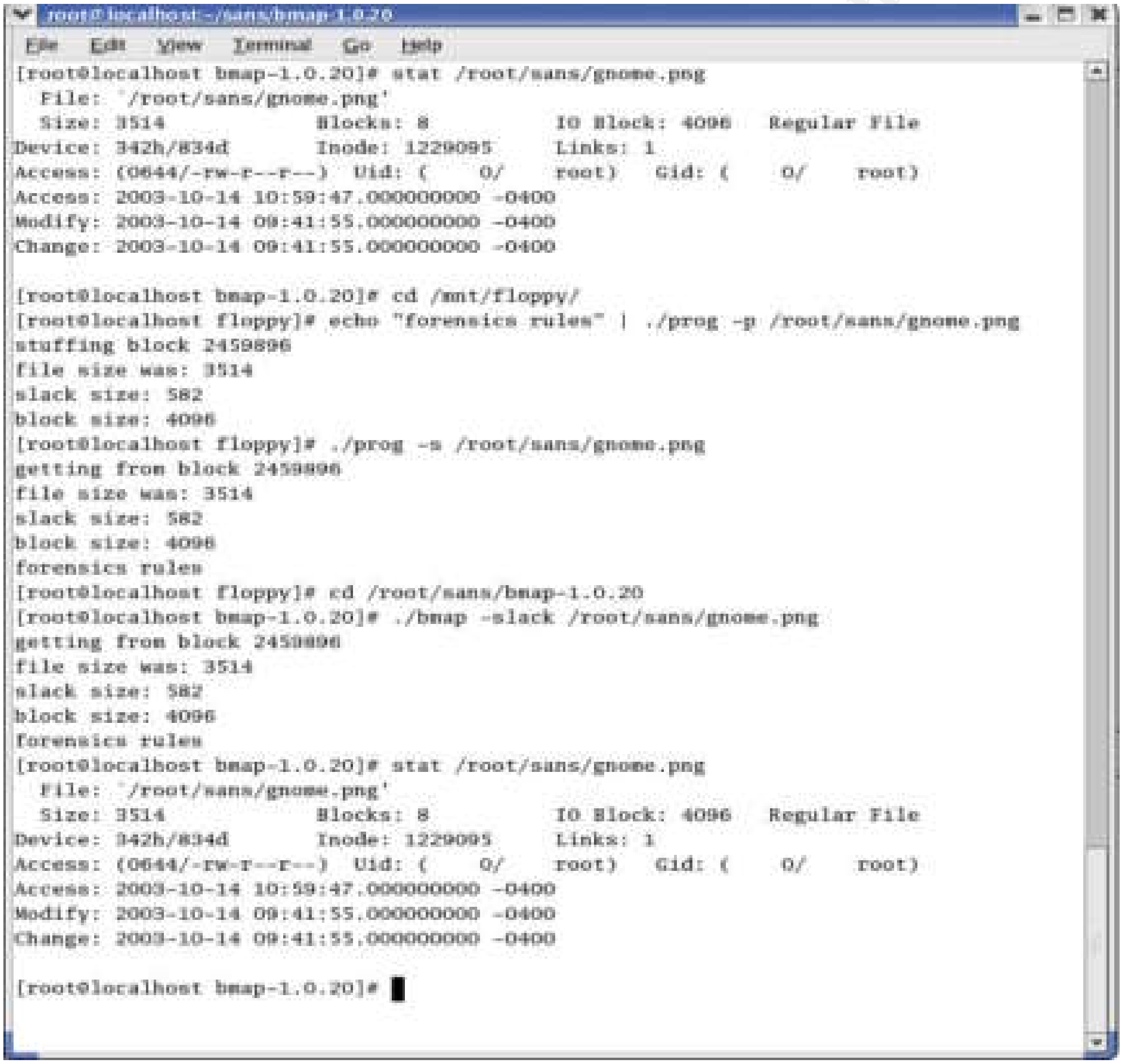

root@localhost:~/sans/bmap-1.0.20
File Edit View Terminal Go Help

```
[root@localhost bmap-1.0.20]# md5sum bmap
91181abd1818b8f73dff2b68596edb07d  bmap
[root@localhost bmap-1.0.20]# ./bmap --help
bmap:1.0.20 (10/14/03) newt@scyld.com
Usage: bmap [OPTION]... [<target-filename>]
use block-list knowledge to perform special operations on files

--doc VALUE
  where VALUE is one of:
  version  display version and exit
  help  display options and exit
  man  generate man page and exit
  sgml  generate SGML invocation info
--mode VALUE
  where VALUE is one of:
  map  list sector numbers
  carve  extract a copy from the raw device
  slack  display data in slack space
  putslack  place data into slack
  wipeslack  wipe slack
  checkslack  test for slack (returns 0 if file has slack)
  slackbytes  print number of slack bytes available
  wipe  wipe the file from the raw device
  frag  display fragmentation information for the file
  checkfrag  test for fragmentation (returns 0 if file is fragmented)
--outfile <filename> write output to ...
--label useless bogus option
--name  useless bogus option
--verbose       be verbose
--log-thresh <none | fatal | error | info | branch | progress | entryexit> logging threshold ...
--target <filename> operate on ...
[root@localhost bmap-1.0.20]# █
```

Figure 1.6

After the bmap file's help menu was reviewed, actual execution of the program was conducted. Figure 1.7 showed the Linux stat command of the gnome.png test file. The stat command gives file size and MAC times of the test file prior to the use of the bmap program. The prog file was used to place data into the slack space of the test file gnome.png and the bmap file was then used to view the data placed into the slack space of the file gnome.png. The words "forensics rules" were placed into the slack with prog and the bmap file was used to view the words placed into the slack. The stat command was run against the gnome.png file again to establish that the bmap program and the prog program do not alter the file size or MAC times of the test file. The commands and the results were captured and displayed in Figure 1.7. No additional forensic footprints were discovered in the analysis of the bmap program which was the same result of the prog file. As the bmap and prog files contain the same usage and function in the same manner, the bmap file was invariably the true name of the prog file.



```
[root@localhost bmap-1.0.20]# stat /root/sans/gnome.png
  File: `/root/sans/gnome.png'
  Size: 3514            Blocks: 8          IO Block: 4096   Regular File
Device: 342h/834d       Inode: 1229095     Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2003-10-14 10:59:47.000000000 -0400
Modify: 2003-10-14 09:41:55.000000000 -0400
Change: 2003-10-14 09:41:55.000000000 -0400

[root@localhost bmap-1.0.20]# cd /mnt/floppy/
[root@localhost floppy]# echo "forensics rules" | ./prog -p /root/sans/gnome.png
stuffing block 2459896
file size was: 3514
slack size: 582
block size: 4096
[root@localhost floppy]# ./prog -s /root/sans/gnome.png
getting from block 2459896
file size was: 3514
slack size: 582
block size: 4096
forensics rules
[root@localhost floppy]# cd /root/sans/bmap-1.0.20
[root@localhost bmap-1.0.20]# ./bmap -slack /root/sans/gnome.png
getting from block 2459896
file size was: 3514
slack size: 582
block size: 4096
forensics rules
[root@localhost bmap-1.0.20]# stat /root/sans/gnome.png
  File: `/root/sans/gnome.png'
  Size: 3514            Blocks: 8          IO Block: 4096   Regular File
Device: 342h/834d       Inode: 1229095     Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2003-10-14 10:59:47.000000000 -0400
Modify: 2003-10-14 09:41:55.000000000 -0400
Change: 2003-10-14 09:41:55.000000000 -0400

[root@localhost bmap-1.0.20]# █
```

Figure 1.7

Legal Implications:

If PRICE downloaded or uploaded MP3 files that were not legally obtained or paid for (with a retail value in excess of $1000), then he is guilty of violating United States Federal Copyright laws in Accordance with Title 17-506 of the

United States Code. The United States Code also states the Copyright laws include civil as well as criminal remedies against offenders. Actual damages as well as statutory damages (up to $30,000) can be obtained. The offender also forfeits his rights to any of the equipment used in the copyright infringements. Based on the floppy disk provided, PRICE had a list of where ripped MP3's are stored. The list was found in slack space which indicates PRICE's desire to hide the information. Log files from the noted sites could be found which document the file transfers of PRICE (home or work computer IP address) to the storage sites. Hiding information is not a crime, it is a right that all computer users have available to them. The hiding of information maybe unauthorized in certain corporate and government offices, but a home user can hide information wherever he or she chooses. The hidden information may be clues to unauthorized or illegal activities (as in PRICE’s case), but it could be something as simple as hiding a phone number or a spouse’s birth date for later retrieval.

Interview Questions:

Assuming the opportunity arose to interview PRICE as a suspect for the installation and execution of the program and assume the interviewer was a Law Enforcement official, the following questions would be presented to PRICE. The varying type of interview techniques would have to be evaluated to determine the best approach; however, in this case, the direct approach would work best.

1. What version of Linux do you use and what are the user and group id's of the accounts?

2. What is your background in the Linux file system?

3. What level/abilities of a Linux user do you consider yourself? Advanced? Novice?

4. How much time do you spend using a Linux computer per day? Per week?

5. Explain what slack space is.

6. What are the methods for storing information in slack space of a file?

7. What are additional methods of placing data in unnoticeable locations (e.g. hidden directories)?

8. What is the process for ripping MP3 files?

9. What internet service provider do you use for your home computer and what is the IP address of the computer(s) you use at work?

10. Have you ever accessed www.fileshares.org, www.convenience-

city.net/main/pub/index.htm, emmpeethrees.com/hidden/index.htm, ripped.net/down/secret.htm

11. Have you ever shared any files (specifically music) over the internet?

12. Are you in possession of any MP3 files legally obtained? Unlawfully obtained?

13. What is your level of access on the network (user, administrator)?

14. Have you received training on the organizational policies for computer use?

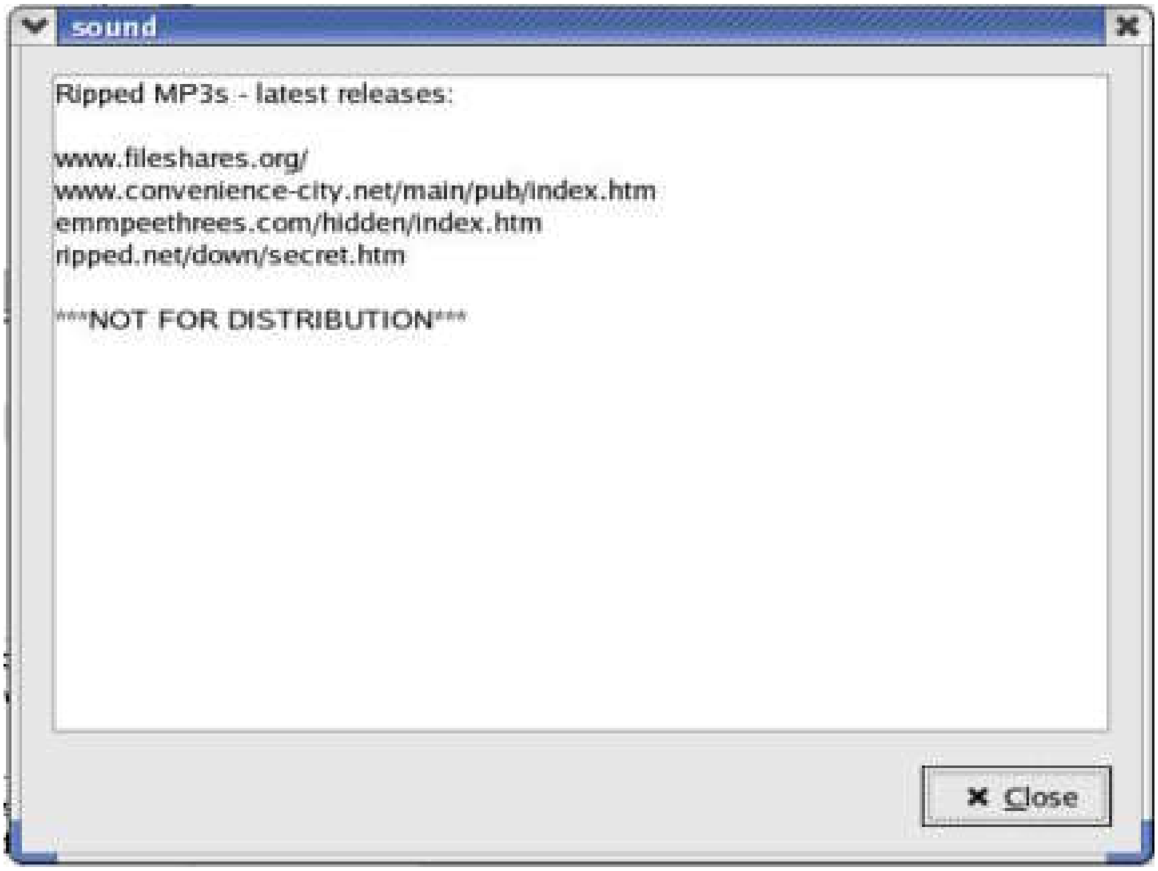
Case Information:

As the prog file can be accessed from external media, such as a floppy disk, then it would be difficult to tell if an individual was using the program. An additional difficulty made for System Administrators is that the program does not alter the file in which the slack space is used. The times and sizes of files are not modified, so even a dutiful administrator who watches over his files would be hard pressed to find the information. A strings or file search of the slack space of a drive (using a tool such as Autopsy) would locate any text in the slack space. Additionally, forensic tools such as Lazarus and Foremost will carve out files from swap space or dd images. Lazarus can retrieve any data from any binary file and group it under different categories, such as text, graphics and C code. Foremost is a free forensics tool created for the Linux platform and developed by Special Agents Kris Kendall and Jesse Kornblum of the U.S. Air Force Office of Special Investigations. The tool was inspired by, and designed to imitate the functionality of, the DOS program CarvThis, written by the Defense Computer Forensics Lab. Foremost enables forensic examiners to automatically recover files or partial files from a bit image. or the media itself based on file header and footer types specified in a user-defined configuration file. Files or text located in slack space would lead an Administrator to believe a user may be utilizing unauthorized tools for storage of their information. If the search of slack space develops text strings or files hidden within several files, then an Administrator should be aware a tool is likely in use. A logical back-up of files or disks would result in the loss of the hidden data as slack space is not a part of the logical file structure. The Administrator may not be able to catch the usage of the tool, but he can prevent its usage by removing the data stored in slack space.

The dd image provided was mounted as read-only to the /mnt/floppy directory of a computer forensic workstation using RedHat Linux 9.0. The dd image was also mounted into the forensic utility Sleuth Kit and the Autopsy program was used to display the results of the Sleuth Kit utilities. The Autopsy program was used to review the additional files found on the floppy disk. All MAC times, UID/GID, strings, and md5sums were obtained using the tools located in Sleuth Kit and viewed with Autopsy.

The floppy disk contained a Docs directory which contained compressed web pages describing how to play DVD movies in Linux (DVD-HOWTO-html.tar.gz), documentation on kernel configuration, compilation, upgrades, and troubleshooting for ix86-based systems (Kernel-HOWTO-html.tar.gz), describing the hardware, software and procedures needed to encode, play, mix and stream MP3 sound files under Linux (Sound-HOWTO-html.tar.gz), and documentation on sound support for Linux (Sound-HOWTO-html.tar.gz). The documentation could be indicative of a person downloading, ripping, and or distributing MP3 files from DVD's and CD's. Copyrighted music can be obtained from a CD and shared across the internet illegally. Each of the tar.gz files possessed the same UID and GID of the prog file (502/502). Based on the nature of the topics covered in the "How-To" documents, the existence and possession of the documents lend credit to the fact PRICE was ripping MP3's likely for distribution.

The prog file was used as a forensic utility against all of the files located on the floppy disk. When the prog command was used in conjunction with the –s switch (show data), an unknown file was discovered. The output was sent to the file sound using the command ./prog –s /Docs/Sound-HOWTO-html.tar.gz >> /root/sans/sound. The Linux command file was used against the sound file, which revealed the file contained gzip compressed data labeled as "downloads". The sound file was renamed to sound.gz and was uncompressed. The uncompressed file revealed hidden text regarding "Ripped MP3's". Figure 1.8 is a screen shot of the text found inside of the sound.gz file. PRICE likely used the prog file to hide the text detailing the location of the actual storage sites of the copyrighted material in the Sound-HOWTO-html.tar.gz file.



Ripped MP3s - latest releases:

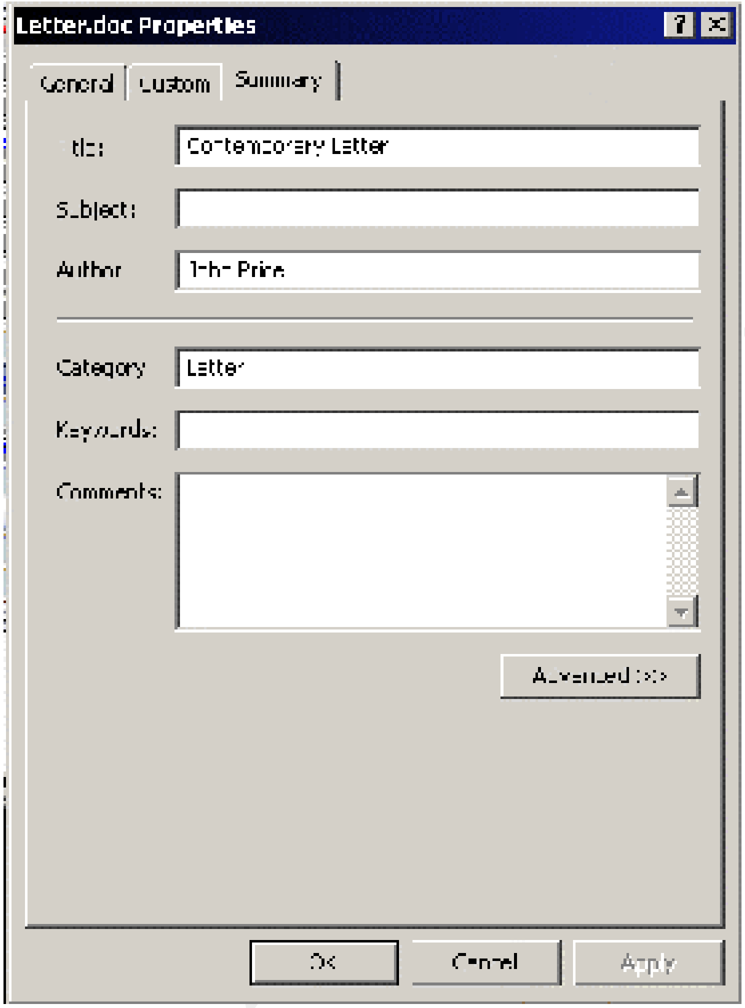
www.fileshares.org/
www.convenience-city.net/main/pub/index.htm
emmpeethrees.com/hidden/index.htm
ripped.net/down/secret.htm

NOT FOR DISTRIBUTION

Figure 1.8

When a file is copied from one disk to another, the slack space is not copied and only the logical file is moved. Slack space which resided at the end of a file on a particular disk will not exist if the file is moved to another disk such as a floppy disk. If data was stored in the slack space of a particular file, a strings command against the disk would display the text that was stored (e.g. credit card numbers, account numbers, etc.).It would be difficult to say if PRICE was able to divide up MP3 files (or any copyrighted material) and store it across slack space, as he would need media larger than a 1.44Mb floppy disk. A search of home or office computers may result in additional files and text stored in slack space.

While PRICE denied the floppy disk belonged to him, the documents contained in the Docs directory (Letter.doc and Mikemsg.doc) show PRICE was the author of the two Microsoft Word documents. The two documents were exported to /root directory on the forensic workstation from the floppy disk image using the Autopsy forensic tool. The two documents were viewed using Microsoft Word XP and the document's properties embedded within the files showed the author was John Price. Figures 1.9 and 1.10 showed the properties view of the Letter.doc and the Mikemsg.doc files.



Title: Contemporary Letter

Subject:

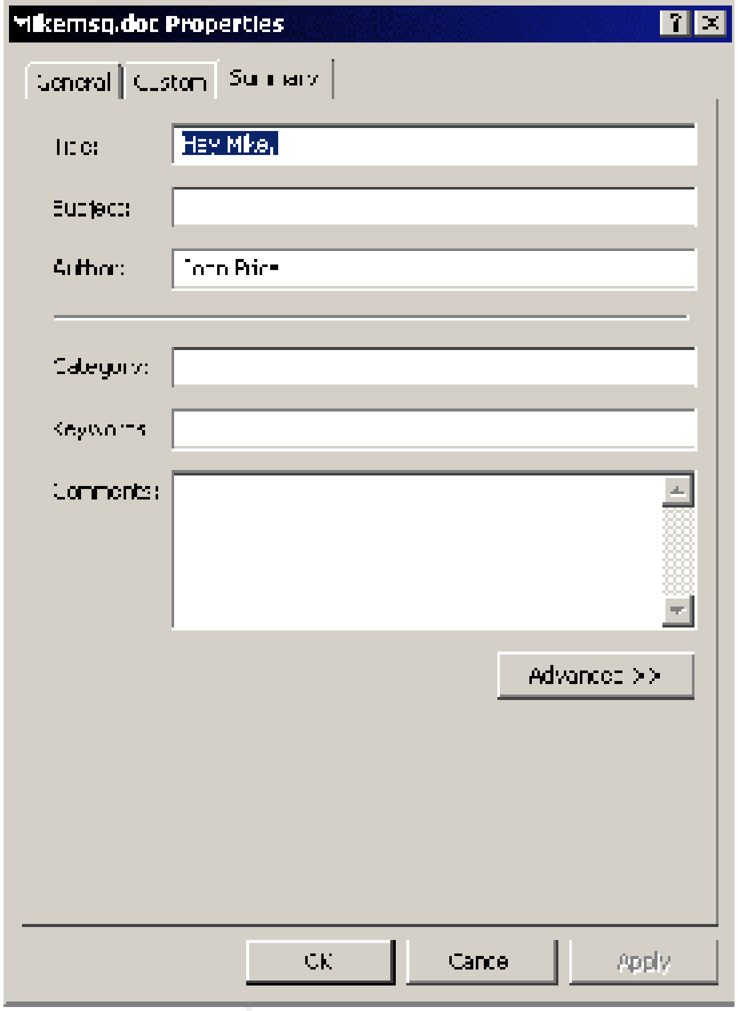
Author: John Prine

Category: Letter

Keywords:

Comments:

Figure 1.9



Mikemsg.doc Properties

Title: Hey Mike,

Subject:

Author: John Price

Category:

Keywords:

Comments:

Figure 1.10

Additional Information:

The SecurityFocus website http://www.securityfocus.com/tools/1359 provided a summary of the bmap program and was also and additional source for downloading a compressed archive of the bmap program. The website http://www.linuxsecurity.com/feature_stories/data-hiding-forensics.html provided an article written by Anton Chuvakin, Ph.D., which detailed the process of data hiding as well as the usage of the bmap program. The website http://www.cs.fsu.edu/~yasinsac/group/slides/busey4.pdf provided a slide show presentation of the technique for hiding files and the use of the bmap program. The bmap program was written by Dan Ridge (newt@scyld.com), Scyld Computing Corporation, Annapolis, Maryland, and would be a likely source for additional information pertaining to the bmap program. Additionally, http://build.lnx-bbc.org/packages/fs/bmap.html

Part II – Forensic Analysis
(Option 1)

Synopsis of Case Facts:

On 15 Jan 03, an incident handler of a Computer Emergency Response Team (CERT) reported the possible compromise of a U.S. Army computer assigned Internet Protocol (IP) address 123.45.67.89, Fully Qualified Domain Name (FQDN) something.army.mil, belonging to the U.S. Army, Fort Dix, NJ. The incident handler believed the computer was compromised as intrusion detection logs of the network revealed file transfer protocol (ftp) traffic to the U.S. Army computer, when the computer should not be hosting the ftp service.

On 16 Jan 03, the system administrator used a java script utility (designed by the U.S. Army Computer Crime Investigative Unit) which collects log files (IIS, event, etc.) and computer status information from the compromised computer. A review of the log collector results revealed the unauthorized and non-standard ports 43958 and 65130 were open and active at the time of the log collection. The number 736 was the unique process identification number for the running program entitled rundll.exe. The rundll.exe program was located in an unauthorized directory, C:\progra~1\Microsoft\Update\dll\tk.

```
736   rundll        43958 TCP   c:\progra~1\Microsoft\Update\DLL\tk\rundll.exe
736   rundll        65130 TCP   c:\progra~1\Microsoft\Update\DLL\tk\rundll.exe
```

Describe the System:

The computer was a workstation using Microsoft Windows 2000 with Service Pack 2, belonging to the Transportation Coordinate Automated Command, Fort Dix, NJ. The computer was not hosting a web page, but did have Microsoft's Internet Information Services (IIS) and FrontPage installed with the Army's standard installation of Windows 2000 Professional and Office 2000. The computer was assigned its IP address (123.45.67.89) on the DIXNET domain using DHCP. The computer was connected to the Army's non-secure network, and was believed to be outside of any firewalls. Unfortunately, the system administrator of the computer could provide no additional relevant information.

Hardware:

On 4 Mar 03, SA CALLAHAN received a 40 GB Western Digital brand IDE drive, Model Number WD400BB-53AUA1, Serial Number WMA6R3666058, from the owner via Certified Mail, Tracking Number 7001 2510 0008 8044 9023, which was contained in the system assigned IP address 123.45.67.89. The hard drive and not the entire system, was forwarded to SA CALLAHAN for examination as standard operating procedure for the U.S. Army. Additional details about the type of equipment used in the system (e.g. processor, CD-ROM, tape drive) was unavailable as the entire system setup was not sent for analysis.

ITEM NO.	QUANTITY	DESCRIPTION OF ARTICLES
0069-1	1	IDE hard drive, Western digital brand, MN: WD400BB-53AUA1, SN: WMA6R3666058, obtained from certified mail #7001 2510 0008 8044 9023
-------	---------	------------------------------LAST ITEM------------------------------

Image media:

On 4 Mar 03, SA CALLAHAN created a digital image of the hard drive from the computer assigned IP address 123.45.67.89 using the Encase® forensic program. EnCase® obtains a bit for bit image of media creating image files that allows the user to view an entire drive including hidden and unallocated space. The program allows you to view files without altering the contents or MAC times. The 40 Gb Western Digital evidence drive was placed into a forensic workstation as the primary slave. The primary master drive was a clean FAT32 formatted 200 Gb hard drive used to store the image files. The forensic workstation was booted using an EnCase® bootable floppy disk which prevents the hard drives from being written to by removing all references to the hard drives (e.g. C:/)as well as using key files located only on the floppy disk (no operating system files). The storage disk was unlocked and the EnCase® images were obtained using EnCase® version 3.22g for DOS. The Encase® image files were seized as evidence and placed on an Evidence/Property Custody Document, Voucher 046-03, and entered into the Evidence Room, this office. Once the images were acquired and verified by Encase® an md5sum hash was created which matched the acquisition hash. This indicates an exact duplicate of the drives data was obtained. The EnCase® acquisition information was provided below and also included the hash values obtained for the acquisition and verification.

Name:	123.45.67.89
File Ext:	117
Description:	Physical Disk, 78165360 Sectors, 37.3GB
Logical Size:	0
Physical Size:	512
Starting Extent:	0S0
Physical Location:	0
Evidence File:	123.45.67.89
Full Path:	0069-02-51.117\123.45.67.89
Device	
Evidence Number:	1
File Path:	Z:\0069-02\123.45.67.89 (046-03)\123.45.67.89.E01
Examiner Name:	Jennie Callahan
Actual Date:	03/04/03 02:47:15PM
Target Date:	03/04/03 02:47:37PM

Total Size:	40,020,664,320 bytes (37.3GB)
Total Sectors:	78,165,360
File Integrity:	Completely Verified, 0 Errors
EnCase Version:	3.22g
System Version:	Windows 2000
Acquisition Hash:	25D1CF021BB9811399C8111C6DE013A7
Verify Hash:	25D1CF021BB9811399C8111C6DE013A7

Media Analysis:

Preliminary Information

Examination of the hard drives was conducted using the Encase® forensic program and the EnCase® image files obtained on 4 Mar 03. The images obtained were exact duplicates and use of the EnCase® program ensures no computer evidence is altered by generating CRC and md5 hash values, which it continuously verifies. EnCase® also allows the user to view an entire drive including hidden and unallocated space from raw or EnCase® images. The program allows you to view files without altering the contents or MAC times. A user can also build and use hash libraries for comparison of known files such as operating system files, rootkits, child pornography. The EnCase® program allows logical and physical searches from regular expressions or text to assist in locating keywords in an examination.

The evaluation of the tk rootkit was a lengthy and time consuming task, which was completed by examinations of over 10 U.S. Army victim computers as well as the computers of the subjects/developers of the rootkit. The rootkit used a known vulnerability in computers running IIS (Primarily Windows 2000 and Professional systems) and installed IRC and FTP software. Once a computer was compromised it would look for additional vulnerable computers (within a given IP address range) and compromise them as well similar to a trojan virus. Additional workups of the rootkit were conducted by the Federal Bureau of Investigation as a part of a joint investigation and Symantec Norton for trojan protection. The examination process began in late 2002 and continues today. The initial examinations established similarly named files such as httpodbc, firedaemon, and MSTaskmgr, which were appearing in each of the victim computer examinations. The EnCase® forensic program provided md5sum hash values for each of the twenty-three suspect files. EnCase® was used again to collect the hash values into a set labeled _tk1_rootkit. When additional victims or subject computers arrived for examination, a hash file comparison, using the created hash set would identify known rootkit files, which would assist in narrowing the scope of the examination.

Additional analysis of files such as Firedaemon.exe, MSTaskmgr.exe, rundll.exe, and wait.com was conducted in a manner similar to what was described in Part 1 of this report. Tools such as Firedaemon.exe and wait.com were freely available on the Internet, which included full descriptions of their capabilities and usage. All unknown files were run in a controlled environment on a forensic workstation using Windows 2000 operating system while the system was monitored for changes. When MSKTaskmgr.exe was executed, it was determined to be the mIRC program. When rundll.exe was executed it was determined to be the Serv-U FTP program.

Apart from the controlled executions of unknown files, all examinations were conducted using the EnCase® forensic program version 4.15 in a Windows 2000 operating system environment.

Section A - Examination of the 13 October 2002 Compromise

1. Examination of the files of the C:\ Volume using the EnCase® forensic program and the hash set _tk1_rootkit revealed numerous files located in two separate places on the volume, which were determined to have matching hash values of known TK rootkit files. The information below detailed the files' MAC times, full path, hash values and hash sets. The identification of known rootkit files using hash values assisted in locating additional unknown files installed by the intruders. This allows for more suspect files to be analyzed which may develop additional clues as to the true identity of the intruders. As stated above, the files with known hash values were analyzed by the U.S. Army, Symantec, and the FBI and were shown to be specific to the tk rootkit. A hash analysis of the files of this or any possible victim computer compared to the hash values of the known tk rootkit files would quickly reveal if the computer was compromised.

The httpodbc.dll file was a trojanized file renamed to a trusted system file, which allowed system level access to a computer utilizing Microsoft Internet Information System.

Full Path	C\Inetpub\Scripts\httpodbc.dll
Hash Value	5aa874cce589b41dff0f3782a8a7f5eb
Hash Set	_tk1_Rootkit
File Created	10/13/02 11:04:53PM
Last Accessed	10/13/02 11:09:10PM
Entry Modified	01/13/03 09:06:14AM

The firedaemon.exe file was a known utility that allowed the installation and execution of any application as a Windows NT/2K/XP service.

Full Path	C\Program Files\Microsoft\Update\DLL\tk\Firedaemon.exe
Hash Value	e3bb90916eb76946eb51f563ffe526f7
Hash Set	_tk1_Rootkit
File Created	10/13/02 11:08:54PM
Last Accessed	01/22/03 07:32:37PM

Entry Modified	01/13/03 09:07:07AM

The MSTaskmgr.exe file was the program mIRC32 renamed to disguise the IRC program while running. The copyright information embedded in the file revealed the file belonged to DOPE Co Ltd.

Full Path	C\Program Files\Microsoft\Update\DLL\tk\MSTaskmgr.exe
Hash Value	2bcb4685df7a34a3ebf9475ee3a45b77
Hash Set	_tk1_Rootkit
File Created	10/13/02 11:08:54PM
Last Accessed	01/22/03 07:32:37PM
Entry Modified	01/13/03 09:07:07AM

The Rundll.exe file was the program Serv-U FTP renamed to disguise the FTP program while running.

Full Path	C\Program Files\Microsoft\Update\DLL\tk\Rundll.exe
Hash Value	814f8e0408a3f6bdfbde6a5d276c6a3a
Hash Set	_tk1_Rootkit
File Created	10/13/02 11:08:54PM
Last Accessed	01/22/03 07:32:40PM
Entry Modified	01/13/03 09:07:07AM

The ServUCert.crt and ServUCert.key files were used in conjunction with the Serv-U FTP program allowing for encryption in the transfer of files.

Full Path	C\Program Files\Microsoft\Update\DLL\tk\ServUCert.crt
Hash Value	9958a2d626a1ea6c21493518f3c5e4b1
Hash Set	_tk1_Rootkit
File Created	10/13/02 11:08:54PM
Last Accessed	01/22/03 07:35:23PM
Entry Modified	01/13/03 09:07:07AM

Full Path	C\Program Files\Microsoft\Update\DLL\tk\ServUCert.key
Hash Value	8226af530ee2f14177f87686a40260f7
Hash Set	_tk1_Rootkit
File Created	10/13/02 11:08:54PM
Last Accessed	01/22/03 07:35:23PM
Entry Modified	01/13/03 09:07:07AM

The wait.com file was a known utility designed to delay the execution of a program for a set time or until user input is received.

Full Path	C\Program Files\Microsoft\Update\DLL\tk\wait.com
Hash Value	8e3f1ab2be0eecbb1e43449861eeb324
Hash Set	_tk1_Rootkit

File Created	10/13/02 11:08:54PM
Last Accessed	10/13/02 11:09:03PM
Entry Modified	01/13/03 09:07:07AM

2. Examination of the C:\Program Files directory revealed four unauthorized subdirectories, Microsoft\Update\DLL\tk, were created at 11:08:54 PM (GMT) [07:08:06 PM (EDT)], 13 October 2002. The tk directory contained 26 unauthorized files. Each of the files was created at the same time as the directory except for three (remote.ini, ServuStartUpLog.txt, and Q269862_W2K_SP2_x86_en.EXE).
    a. Examination of the tk00.tmp file revealed it was the mIRC configuration for the U.S. Army computer while running the mIRC program. The file enabled the U.S. Army computer to connect to an IRC server (irc.infatech.net) on port 6667 as "vMz_r00t_685" Figure 2.1a was the contents of the tk00.tmp file.

```
[mirc]
user=TK
email=TK@fbi.gov
nick=vMz_r00t_685
anick=TK-ALT
host=irc.infatech.netSERVER:irc.infatech.net:6667
```

Figure 2.1a

    b. Examination of the tk.conf file revealed it was a bot configuration file for the U.S. Army computer while running the IRC program. The file enabled the U.S. Army computer to run as a "bot" on IRC using port 4323. The bot used the servers "irc.infatech.net" or "ware123.mine.nu" (srv.cnf) on channel #vMz-Bots ownz (jn.cnf). The IRC bot was started at 11:08:59 PM (GMT), 13 October 2002. Figure 2.1b was the contents of the tk.conf file.

```
[settings]
version=TKbot.R00t.EDITiON.FiNAL
version.msg=0wnz j00 4LL
tag=vMz_r00t
bport=4323
bpwd=0wnzj00
busers=5
pwd=*"þ+=;ß
cmode.s=on
cmode=+ts
rand=on
nick=TK-BOT
serverfile=srv.cnf
```

```
channelfile=jn.cnf
ipfile=i.p
su=Sun Oct 13 19:08:59 2002
upt=944799
onl=2
```

Figure 2.1b

c. Examination of the su.txt and suw.txt files revealed the files were used as welcome screens when a user logged on to the Serv-U FTP server.

d. Examination of the servudaemon.ini file revealed the file was a configuration file for the Serv-U FTP program. Figure 2.1d was the contents of the servudaemon.ini file.

```
[GLOBAL]
Version=4.0.0.4
ProcessID=740
RegistrationKey=AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAagAAA6mMGTxCDgL/C2
p1YW4gdmFsZGV6BXl1YmFu
AntiHammer=1
AntiHammerWindow=60
AntiHammerTries=2
PacketTimeOut=300

[DOMAINS]
Domain1=0.0.0.0||65130|TK|1|1

[Domain1]
ReplyHello=###     TK Distro     ###
ReplyTooMany=Too many users - dont hammer me
User1=d00m|1|0
LogSystemMes=0
LogSecurityMes=0
LogGETs=0
LogPUTs=0
LogFileSystemMes=0
LogFileSecurityMes=0
LogFileGETs=0
LogFilePUTs=0
SignOn=c:\progra~1\Microsoft\Update\DLL\tk\su.txt
DirChangeMesFile=c:\progra~1\Microsoft\Update\DLL\tk\suw.txt
MaxNrUsers=10
User2=tk|1|0

[USER=tk|1]
Password=hrA4BB52020629001C3FA88C701832B89D
HomeDir=c:\winnt\system32\shellext\system\tk
RelPaths=1
TimeOut=600
MaxNrUsers=5
Access1=c:\winnt\system32\ShellExt\System\TK|RWALCDP
```

Serv-U FTP program valid registration key

Serv-U FTP program user names and passwords with the user directory of the hidden directory installed at 11:09:06 PM (GMT), 13 October 2002

```
[USER=d00m|1]
Password=mc69C727139CC5EDD21A166C73501B1CD2
HomeDir=c:\winnt\system32\shellext\system\tk
AlwaysAllowLogin=1
TimeOut=600
Maintenance=System
Access1=c:\|RWAMELCDP
Access2=d:\|RWAMELCDP
Access3=e:\|RWAMELCDP
Access4=f:\|RWAMELCDP
```

Figure 2.1d

e. Examination of the ServUCert.key & ServUCert.crt files revealed they are the Secure Sockets Layer (SSL) Certificate and the Serv-U FTP server’s private key used for secure (encrypted) file transfers.

f. Examination of the file rs.exe revealed it was a tool used to change permissions of other files. The tool removed the user associated with the file, and only the “System” user can read or execute the file, subsequently the file could not be deleted.

g. Examination of the libeay32.dll, ssleay32.dll, j.dll and Tzolibr.dll files revealed they were library files required to run the Serv-U FTP or the mIRC programs.

h. Examination of the ms.vxd and crk.vxd files revealed they were Windows registry files which enabled Rundll.exe and MSTaskmger.exe to run at startup as legitimate services. Additionally, the registry value was added to enable mIRC as a registered product with the user name and license of Double A. Ron, 8557-785634.

i. Examination of the d.exe file revealed it was a renamed version of the DOS command deltree, which deletes files to include directories and subdirectories recursively from a computer.

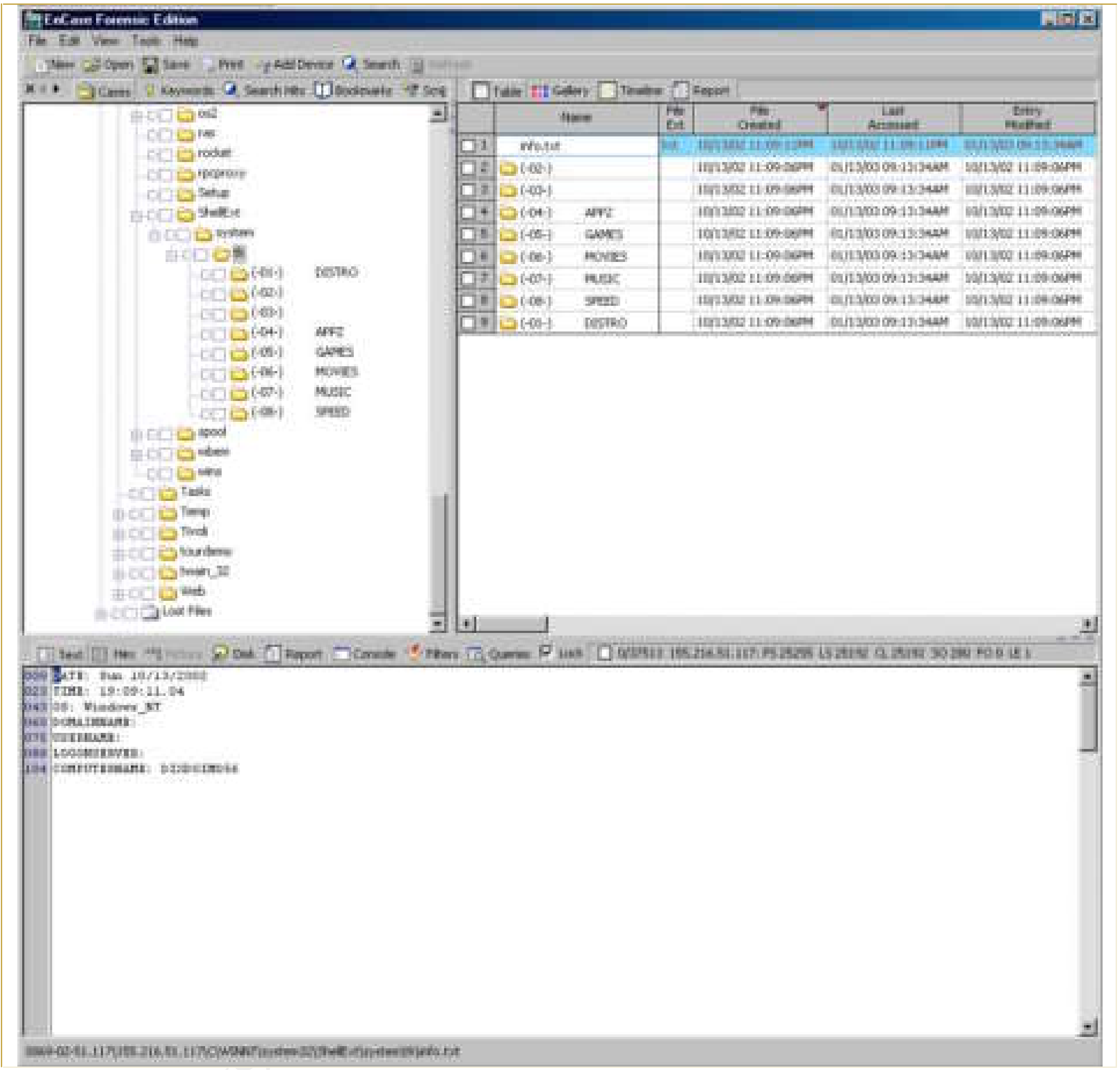
j. Examination of the Q269862_W2K_SP2_x86_en.EXE file revealed it was created at 11:10:40 PM (GMT), 13 October 2002. The file was the Microsoft patch for the vulnerability used by the intruder, but the patch was not installed.

k. Examination of the ServUStartUpLog.txt file revealed it was created at 11:09:01 PM (GMT), 13 October 2002. The file denoted the Serv-U FTP program was last started at 14:35:23 PM (EST), 22 January 2003 on ports 65130 and 43958. Figure 2.1k was the contents of the ServUStartUpLog.txt file.

```
Wed 22Jan03 14:35:20 - Serv-U FTP Server v4.0 (4.0.0.4) - Copyright (c) 1995-2002 Cat Soft, All Rights
Reserved - by Rob Beckers
Wed 22Jan03 14:35:20 - Cat Soft is an affiliate of Rhino Software, Inc.
Wed 22Jan03 14:35:20 - Using WinSock 2.0 - max. 32767 sockets
Wed 22Jan03 14:35:20 - Starting FTP Server...
Wed 22Jan03 14:35:23 - Loaded SSL/TLS libraries
Wed 22Jan03 14:35:23 - FTP Server listening on port number 65130, IP 155.216.51.33, 127.0.0.1
Wed 22Jan03 14:35:23 - FTP Server listening on port number 43958, IP 127.0.0.1
Wed 22Jan03 14:35:23 - Valid registration key found
```

Figure 2.1k

3. Examination of the C:\WINNT\system32\ShellExt\system directory revealed a subdirectory labeled “tk” was created at 11:09:06 PM (GMT), 13 October 2002. The directory contained eight additional directories which were empty, but created at the same time. Figure 3.1 provided a screen shot of the EnCase® forensic program highlighting the subdirectories. The file info.txt was located in the tk directory and contained the U.S. Army computer information.



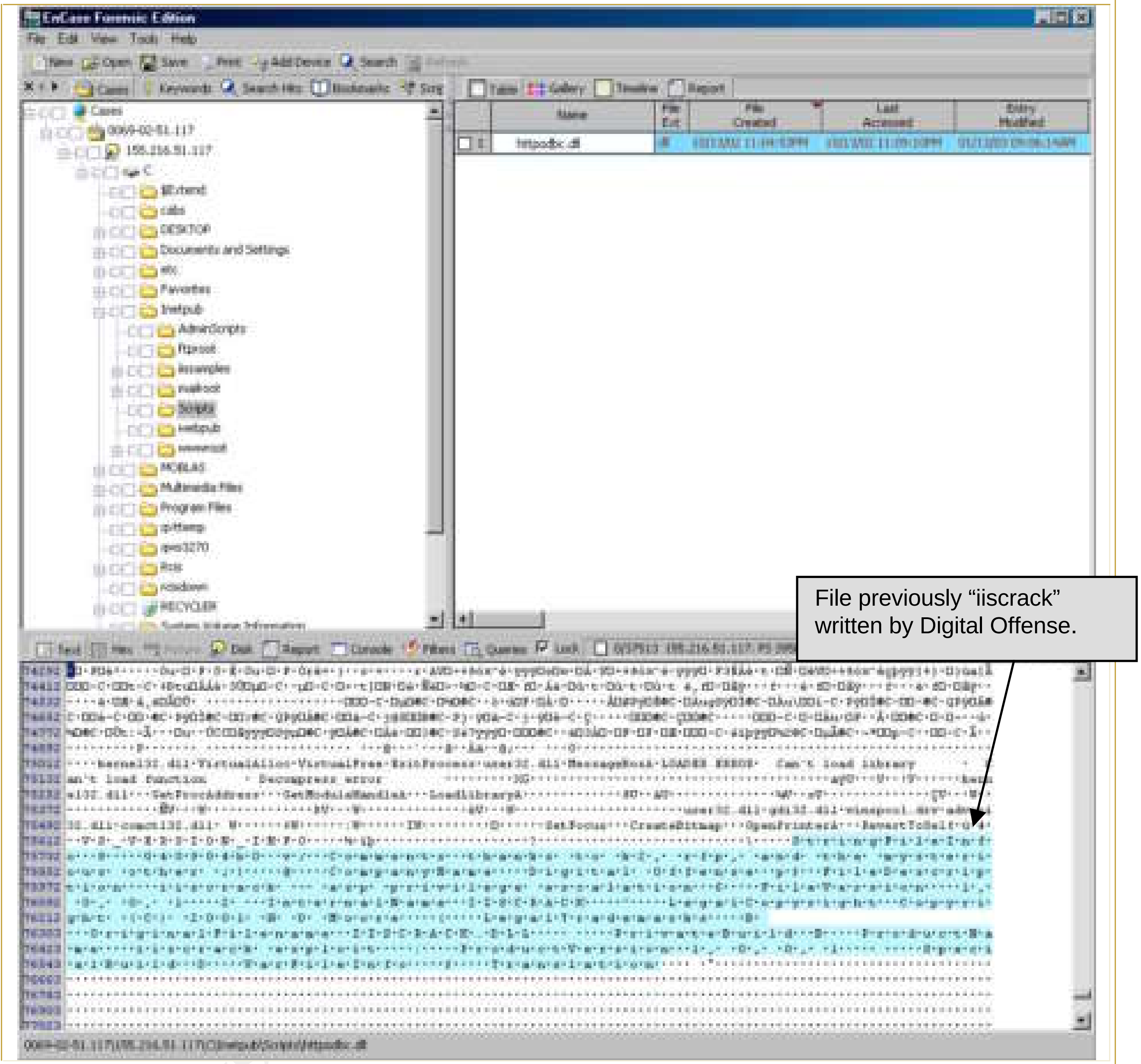
	Name	File Ext	File Created	Last Accessed	Entry Modified
1	info.txt	txt	10/13/02 11:09:11PM	10/13/02 11:09:11PM	01/13/03 09:13:34AM
2	(-02-)		10/13/02 11:09:06PM	01/13/03 09:13:34AM	10/13/02 11:09:06PM
3	(-03-)		10/13/02 11:09:06PM	01/13/03 09:13:34AM	10/13/02 11:09:06PM
4	(-04-) APPZ		10/13/02 11:09:06PM	01/13/03 09:13:34AM	10/13/02 11:09:06PM
5	(-05-) GAMES		10/13/02 11:09:06PM	01/13/03 09:13:34AM	10/13/02 11:09:06PM
6	(-06-) MOVIES		10/13/02 11:09:06PM	01/13/03 09:13:34AM	10/13/02 11:09:06PM
7	(-07-) MUSIC		10/13/02 11:09:06PM	01/13/03 09:13:34AM	10/13/02 11:09:06PM
8	(-08-) SPEED		10/13/02 11:09:06PM	01/13/03 09:13:34AM	10/13/02 11:09:06PM
9	(-01-) DISTRO		10/13/02 11:09:06PM	01/13/03 09:13:34AM	10/13/02 11:09:06PM

```
000 DATE: Sun 10/13/2002
023 TIME: 19:09:11.04
043 OS: Windows_NT
060 DOMAINNAME:
075 USERNAME:
088 LOGONSERVER:
104 COMPUTERNAME: [illegible]
```

0069-02-51.117\195.216.51.117\C\WINNT\system32\ShellExt\system\®\info.txt

Figure 3.1

4. Examination of the C:\Inetpubs\scripts directory revealed one file, httpodbc.dll, which was created at 11:04:53 PM (GMT), 13 October 2002. The file was a trojanized file renamed to a trusted system file, which allowed system level access to a computer utilizing IIS when placed in the \Inetpub\scripts directory. Figure 4.1 showed the text embedded into the end of the httpodbc.dll file as viewed by the EnCase® forensic program.


EnCase Forensic Edition
File Edit View Tools Help
New Open Save Print Add Device Search Refresh
Cases Keywords Search Hits Bookmarks
Table Gallery Timeline Report
Name
File Ext
File Created
Last Accessed
Entry Modified
httpodbc.dll
Cases
0069-02-51.117
155.216.51.117
C
$Extend
cabs
DESKTOP
Documents and Settings
etc
Favorites
Inetpub
AdminScripts
ftproot
iissamples
mailroot
Scripts
webpub
wwwroot
MOBLAS
Multimedia Files
Program Files
qttemp
qws3270
Rcis
rcisdown
RECYCLER
System Volume Information
Text Hex Picture Disk Report Console Filters Queries Lock
kernel32.dll VirtualAlloc VirtualFree ExitProcess user32.dll MessageBoxA LOADER ERROR Can't load library
an't load function Decompress error
GetProcAddress GetModuleHandleA LoadLibraryA
user32.dll gdi32.dll winspool.drv
comdlg32.dll SetFocus CreateBitmap OpenPrinterA RevertToSelf
0069-02-51.117\155.216.51.117\C\Inetpub\Scripts\httpodbc.dll

File previously "iiscrack" written by Digital Offense.

Figure 4.1

5. Examination of IIS Log File ex021013.log revealed the computer assigned IP address 24.148.63.209 was the only computer which connected to the U.S. Army computer and accessed the trojanized file, httpodbc.dll. Again, the computer was not designed to host a web page, although the default service was installed and running on the computer making the computer vulnerable to an IIS based attack. Figure 5.1 showed the content of the log file. The IP address 24.148.63.209 was registered to RCN Corporation, 105 Carnegie Center, Princeton, NJ 08540.

```
Full Path:          C:\WINNT\system32\LogFiles\W3SVC1\ex021013.log
File Created:       10/13/02 11:09:10PM
Last Accessed:      10/13/02 11:09:10PM
Last Written:       10/14/02 12:00:00AM

#Software: Microsoft Internet Information Services 5.0
#Version: 1.0
#Date: 2002-10-13 23:09:10
#Fields: time c-ip cs-method cs-uri-stem sc-status
23:09:10 24.148.63.209 GET /scripts/httpodbc.dll 200
```

File created, accessed, modified and the log file entries were noted as GMT.

Response 200 – Request Successful

Figure 5.1

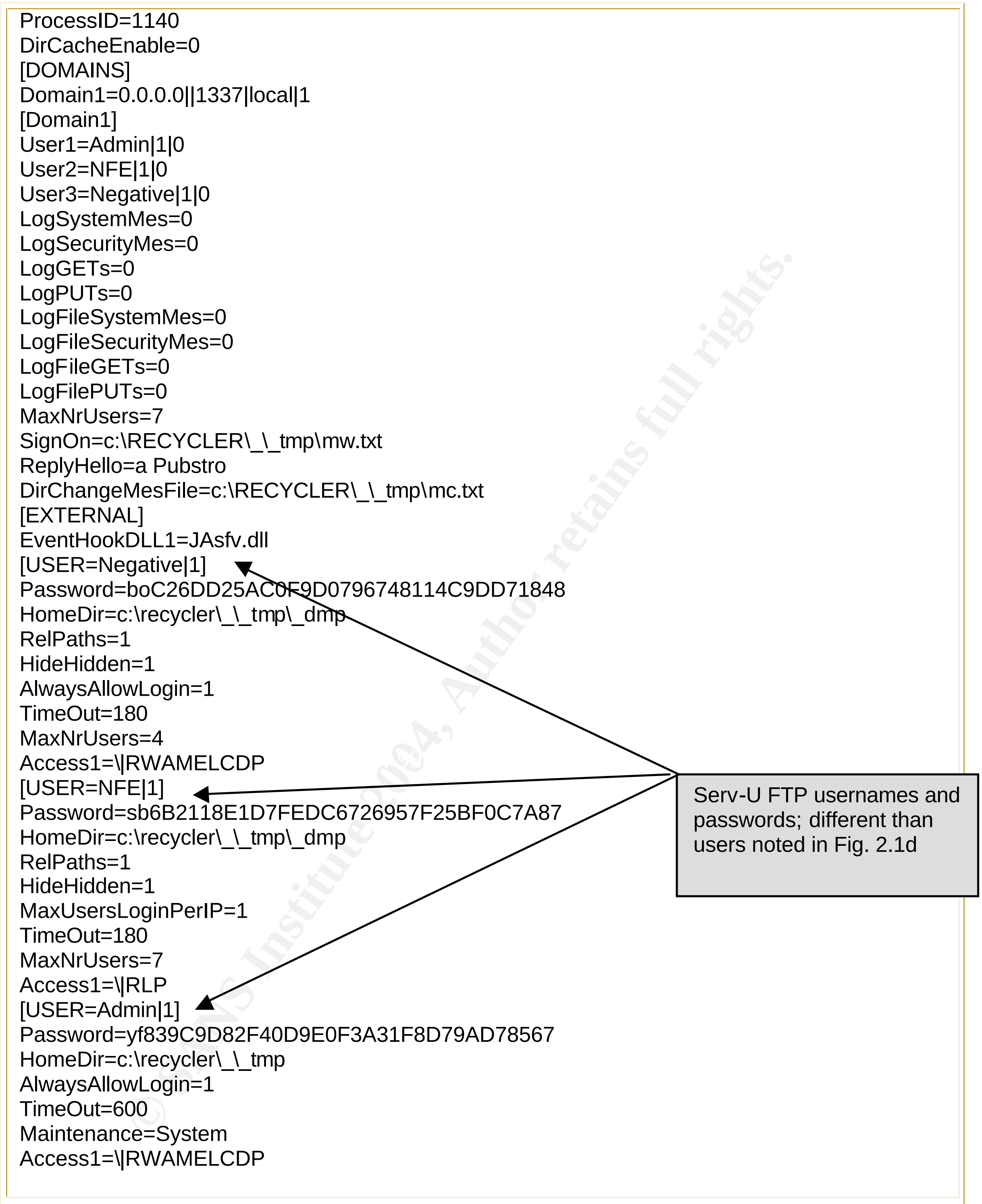
Section B - Examination of the 31 October 2002 Compromise

6. Examination of the C:\WINNT\Temp\ directory revealed three unauthorized subdirectories, recycler\tmp\.tmp, were created at 10:00:19 PM (GMT), 31 October 2002. The recycler\tmp directory contained one unauthorized file.
   a. Examination of the cnd.exe file revealed it was created at 10:00:28 PM (GMT), 28 October 2002. The file was a renamed version of the Windows system file cmd.exe designed to open a Windows command prompt.

7. Examination of the C:\Recycler directory revealed three unauthorized subdirectories, __tmp\dmp, were created at 10:02:25 PM (GMT), 31 October 2002. The __tmp directory contained seven unauthorized files.
   a. Examination of the cnd.exe file revealed it was created at 10:02:28 PM (GMT), 31 October 2002. The file maintained the same hash value as the file noted in Paragraph 10.2.1.1.
   b. Examination of the winmgnt.exe file revealed it was created at10:04:04 PM (GMT), 31 October 2002. The file was a renamed version of the Serv-U FTP program.
   c. Examination of the servudaemon.ini file revealed it was created at10:04:09 PM (GMT), 31 October 2002. The file was a configuration file for the Serv-U FTP program. Figure 7.1 showed the contents of the servudaemon.ini file.

```
[GLOBAL]
Version=3.0.0.17
RegistrationKey=AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAASgAAA
0XVDzwQJwIQCFJlYUxJc1R5BFJlYUw=
BlockAnti TimeOut=1
AntiHammer=1
AntiHammerBlock=1200
PacketTimeOut=300
MaxNrUsers=7
```

Serv-U FTP program valid registration key; different than key noted in Fig. 2.1d



```
ProcessID=1140
DirCacheEnable=0
[DOMAINS]
Domain1=0.0.0.0||1337|local|1
[Domain1]
User1=Admin|1|0
User2=NFE|1|0
User3=Negative|1|0
LogSystemMes=0
LogSecurityMes=0
LogGETs=0
LogPUTs=0
LogFileSystemMes=0
LogFileSecurityMes=0
LogFileGETs=0
LogFilePUTs=0
MaxNrUsers=7
SignOn=c:\RECYCLER\_\_tmp\mw.txt
ReplyHello=a Pubstro
DirChangeMesFile=c:\RECYCLER\_\_tmp\mc.txt
[EXTERNAL]
EventHookDLL1=JAsfv.dll
[USER=Negative|1]
Password=boC26DD25AC0F9D0796748114C9DD71848
HomeDir=c:\recycler\_\_tmp\_dmp
RelPaths=1
HideHidden=1
AlwaysAllowLogin=1
TimeOut=180
MaxNrUsers=4
Access1=\|RWAMELCDP
[USER=NFE|1]
Password=sb6B2118E1D7FEDC6726957F25BF0C7A87
HomeDir=c:\recycler\_\_tmp\_dmp
RelPaths=1
HideHidden=1
MaxUsersLoginPerIP=1
TimeOut=180
MaxNrUsers=7
Access1=\|RLP
[USER=Admin|1]
Password=yf839C9D82F40D9E0F3A31F8D79AD78567
HomeDir=c:\recycler\_\_tmp
AlwaysAllowLogin=1
TimeOut=600
Maintenance=System
Access1=\|RWAMELCDP
```

Serv-U FTP usernames and passwords; different than users noted in Fig. 2.1d

Figure 7.1

d. Examination of the jasfv.dll and jasfv.ini files revealed they were created at10:04:10 and 10:04:12 PM (GMT), 31 October 2002. The files were necessary program files for the Serv-U FTP program.

e. Examination of the mw.txt and mc.txt files revealed they were created at10:04:12 and 10:04:13 PM (GMT), 31 October 2002. The files were used as welcome screens when a user logged on to the Serv-U FTP server.

8. Examination of IIS Log File ex021031.log revealed the computer assigned IP address 213.51.1.133 was the only computer which connected to the U.S. Army computer and accessed the windows command prompt and hidden directory and files. Figure 8.1 showed the contents of the IIS log file, and the commands used by the intruder to gain access. The IP address 213.51.1.133 was registered @Home Benelux Network Operations Centre, Gyroscoopweg 90-92, 1042 AX Amsterdam, Netherlands.

```
Full Path:          C:\WINNT\system32\LogFiles\W3SVC1\ex021031.log
File Created:       10/31/02 07:15:39PM
Last Accessed:      12/23/02 06:28:09PM
Last Written:       11/01/02 12:00:00AM

#Software: Microsoft Internet Information Services 5.0
#Version: 1.0
#Date: 2002-10-31 19:15:39
#Fields: time c-ip cs-method cs-uri-stem sc-status
19:15:39 211.101.209.188 GET / 400
22:00:00 213.51.1.133 GET /winnt/system32/cmd.exe 404
22:00:00 213.51.1.133 GET /scripts/.%2e/.%2e/winnt/system32/cmd.exe 200
22:00:00 213.51.1.133 GET /scripts/.%2e/.%2e/winnt/system32/cmd.exe 200
22:00:01 213.51.1.133 GET /scripts/.%2e/.%2e/winnt/system32/cmd.exe 200
22:00:01 213.51.1.133 GET /scripts/.%2e/.%2e/winnt/system32/cmd.exe 502
22:00:01 213.51.1.133 GET /scripts/.%2e/.%2e/winnt/system32/cmd.exe 502
22:00:15 213.51.1.133 GET /scripts/..%5c..%5cwinnt/system32/cmd.exe 200
22:00:19 213.51.1.133 GET /scripts/..%5c..%5cwinnt/system32/cmd.exe 502
22:00:23 213.51.1.133 GET /scripts/..%5c..%5cwinnt/system32/cmd.exe 500
22:00:28 213.51.1.133 GET /scripts/..%5c..%5cwinnt/system32/cmd.exe 502
22:00:35 213.51.1.133 GET /scripts/..%5c..%5cwinnt/temp/recycler/temp/cnd.exe 502
22:00:42 213.51.1.133 GET /scripts/..%5c..%5cwinnt/temp/recycler/temp/cnd.exe 502
22:00:45 213.51.1.133 GET /scripts/..%5c..%5cwinnt/temp/recycler/temp/cnd.exe 502
22:01:07 213.51.1.133 GET /scripts/..%5c..%5cwinnt/system32/cmd.exe 502
22:01:20 213.51.1.133 GET /scripts/..%5c..%5cwinnt/system32/cmd.exe 200
22:02:22 213.51.1.133 GET /scripts/..%5c..%5cwinnt/system32/cmd.exe 200
22:02:25 213.51.1.133 GET /scripts/..%5c..%5cwinnt/system32/cmd.exe 502
22:02:28 213.51.1.133 GET /scripts/..%5c..%5cwinnt/system32/cmd.exe 502
22:02:54 213.51.1.133 GET /scripts/..%5c..%5crecycler/_/_tmp/cnd.exe 502
22:03:00 213.51.1.133 GET /scripts/..%5c..%5crecycler/_/_tmp/cnd.exe 502
22:03:04 213.51.1.133 GET /scripts/..%5c..%5crecycler/_/_tmp/cnd.exe 502
22:04:14 213.51.1.133 GET /scripts/..%5c..%5crecycler/_/_tmp/cnd.exe 502
22:04:21 213.51.1.133 GET /scripts/..%5c..%5crecycler/_/_tmp/cnd.exe 502
22:04:25 213.51.1.133 GET /scripts/..%5c..%5crecycler/_/_tmp/cnd.exe 502
22:05:21 213.51.1.133 GET /scripts/..%5c..%5cwinnt/system32/cmd.exe 200
22:05:27 213.51.1.133 GET /scripts/..%5c..%5cwinnt/system32/cmd.exe 200
22:05:32 213.51.1.133 GET /scripts/..%5c..%5cwinnt/system32/cmd.exe 200
22:06:10 213.51.1.133 GET /scripts/..%5c..%5cwinnt/system32/cmd.exe 502
22:06:23 213.51.1.133 GET /scripts/..%5c..%5cwinnt/system32/cmd.exe 502
```

File created, accessed, modified times and log file entries were noted as GMT.

Response 200 – Request Successful
Response 502 – Webserver received invalid response

```
22:06:37 213.51.1.133 GET /scripts/..%5c..%5cwinnt/system32/cmd.exe 502
22:06:40 213.51.1.133 GET /scripts/..%5c..%5cwinnt/system32/cmd.exe 502
```

Figure 8.1

Section C - Examination of the Event Logs

9. Examination of the system, security, and application logs was conducted by review of the initial log collector results obtained. Additionally, the event files were exported from the hard drive image using EnCase® and were examined again by importing the log files into the event viewer of a forensic workstation using the Windows 2000 operating system.

10. Examination of the system event logs revealed the Windows protected files, c:\winnt\system32\ftp.exe and tftp.exe, were replaced with unauthorized files and then the original versions were restored at 11:09:11 PM (GMT), 13 October 2002 by the operating system.

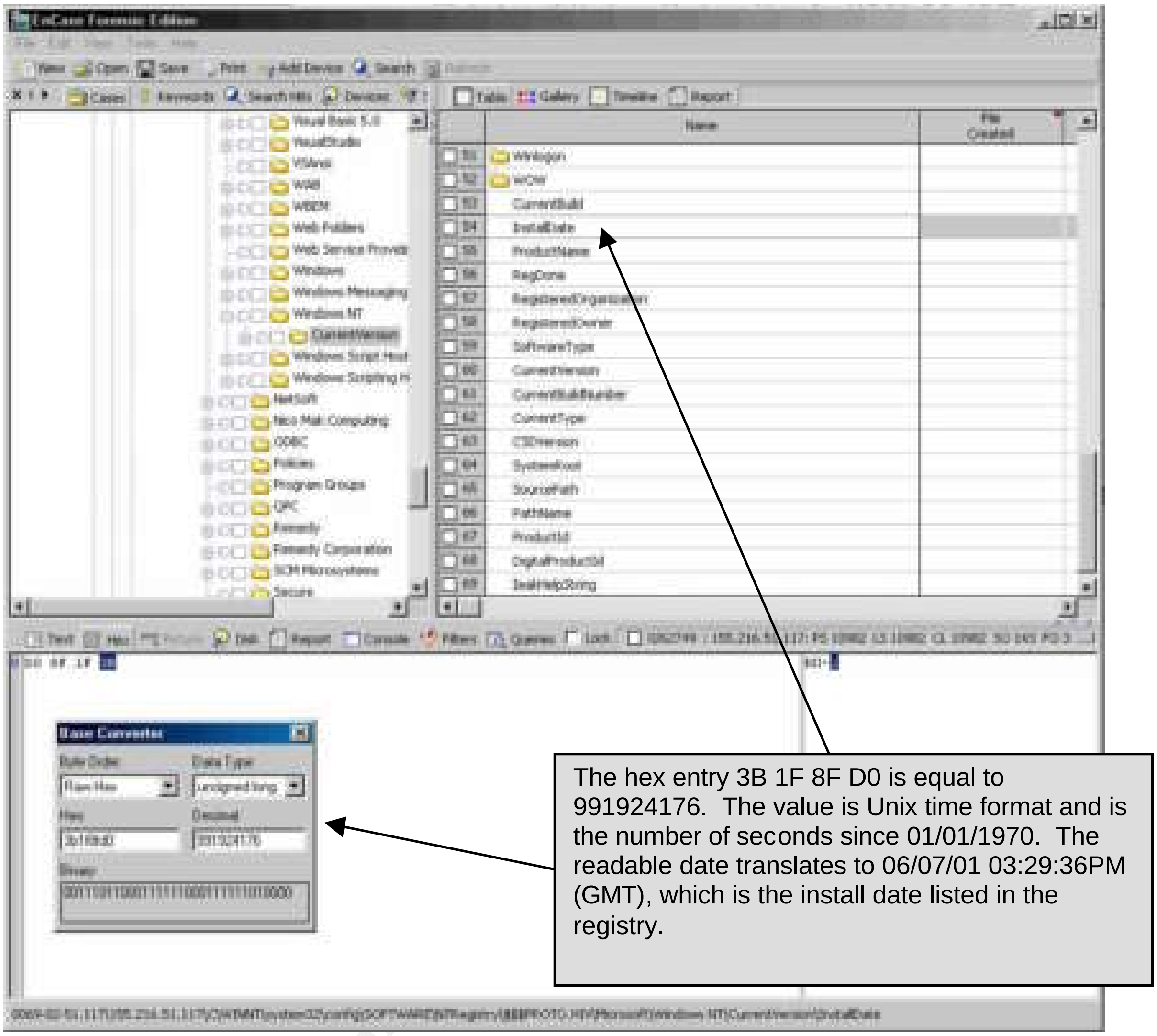
11. Examination of the application event logs revealed the unauthorized services MSTaskMgr and Rundll were active on the U.S. Army computer between 21 October 2002 through 21 January 2003. No application logs were available prior to 19 October 2002.

12. Examination of the security event logs revealed the anonymous internet user (IUSR_DIXDOIM056) had successful logon and logoff events on 11:04:42 PM (GMT) and 11:26:33 PM (GMT), 13 October 2002 and at 10:00:00 PM (GMT) and 10:27:39 (GMT), 31 October 2002.

Timeline Analysis:

EnCase® allows the user to mount registry files in the EnCase® so as the registry can be navigated in the same fashion as other folder structures. The Windows 2000 registry files “system” and “software” were mounted and the installation information was obtained. The Windows Registry information revealed the following pertaining to the installation of the operating system: Install Date: 06/07/01 03:29:36PM (GMT), Product Name: Microsoft Windows 2000, Registered Organization: FT DIX NJ, Registered Owner: US ARMY, Current Version: 5.0, Current Build Number: 2195, CSD Version: Service Pack 2, System Root: C:\WINNT, Source Path: D:\I386, Path Name: C:\WINNT, Product Id: 51873-OEM-0003774-18690, and the Last Shutdown time was at 02:53:05PM (GMT), 22 January 2003

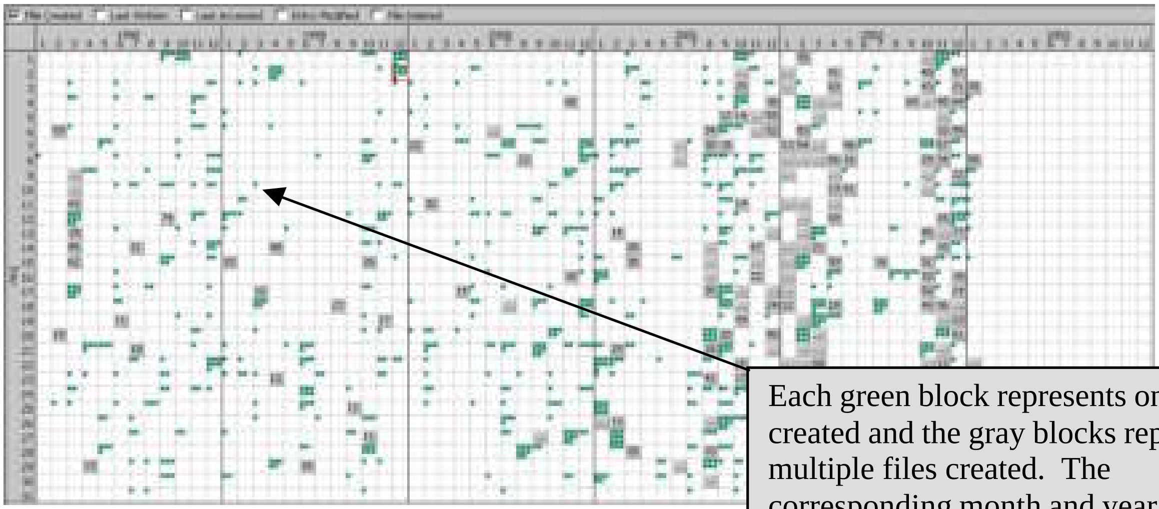
The following EnCase® screen shot showed the registry view as the user would see it after mounting the “software” registry file.



Byte Order: Raw Hex
Data Type: unsigned long
Hex: 3b1f8fd0
Decimal: 991924176
Binary: 00111011000111111000111111010000

The hex entry 3B 1F 8F D0 is equal to 991924176. The value is Unix time format and is the number of seconds since 01/01/1970. The readable date translates to 06/07/01 03:29:36PM (GMT), which is the install date listed in the registry.

The following timeline of all of the files on the computer was obtained using EnCase®. The created, written, accessed, modified and deleted times are shown in color coded squares. Each colored square represents an individual file in its corresponding time or date stamp block on a chart. When more than nine files have the same corresponding time or date, then a gray block showing the number of files of that date stamp is shown until a more detailed view is chosen (e.g. by hour of a day). The figure below represents a table generated for created times of all of the files on the hard drive. A screen similar to the one detailed below can be accessed for each of the written, accessed, modified, and deleted times for selected or all files. A timeline generated in this fashion is difficult to read for all files on the hard drive, but is useful for a graphical representation of the MAC times on a specific file or only a select few files. EnCase® also generates a report documenting each of the computers files and subsequent time and date stamps which is provided. See Jennie_Callahan_GCFA_Timeline.rtf (The document is 4475 pages in length).



Each green block represents one file created and the gray blocks represent multiple files created. The corresponding month and year are located along the top of the chart and the corresponding day runs along the

Created

In an effort to confirm the installation date of 6 July 2001, a review of the core system files was conducted. The key registry files in Windows 2000, SOFTWARE, SECURITY, and SYSTEM, were all noted as being created on 6 July 2001. Additionally, the security event log was first created on 6 July 2001. Although file date and times may be modified, core system files such as the registry are not likely modified as they do not alter relevant information and may likely damage the computer.

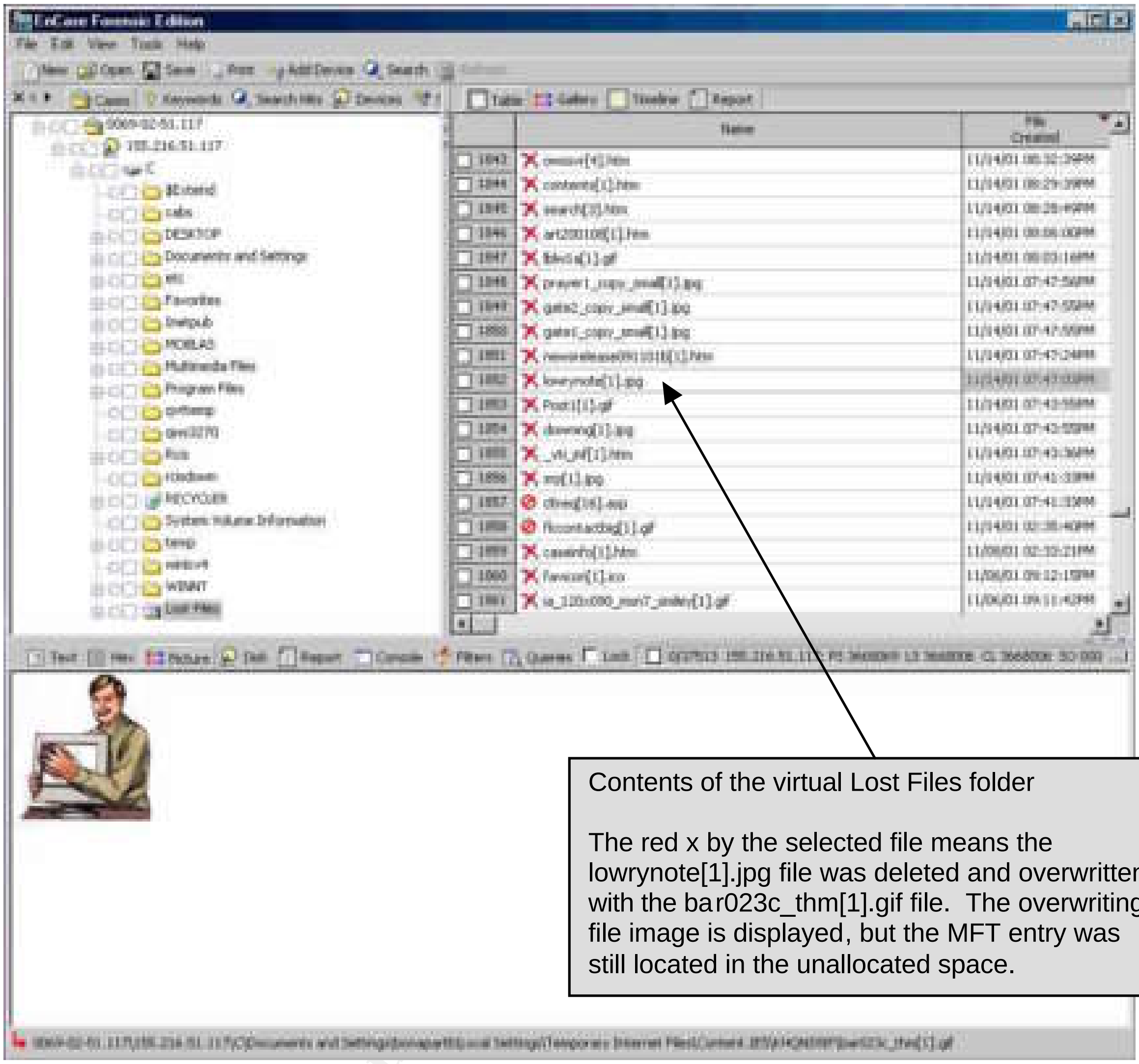
Full Path	C\WINNT\system32\config\SecEvent.Evt
File Created	06/07/01 06:39:54PM
Full Path	C\WINNT\system32\config\SOFTWARE
File Created	06/07/01 02:13:17PM
Full Path	C\WINNT\system32\config\SECURITY
File Created	06/07/01 06:14:11PM
Full Path	C\WINNT\system32\config\SYSTEM
File Created	06/07/01 02:13:17PM

Apart from the previously mentioned files in Media Analysis, no files were noted as relevant to the investigation.

Recover Deleted Files:

Files deleted from a computer by the user are typically moved to the Recycled folder on the drive from which they are deleted. The file actually stays in the same place but its directory entry is removed and placed in the Recycled folder.

The file is then renamed and the original name and location of the file are stored in the index file, called INFO or INFO2 located in the Recycled folder. To restore files, the original path is read from the INFO file, and the file is again renamed and restored to its original path. When files are "emptied" from the Recycled folder, Windows changes the file's directory entry to indicate the space occupied by the file is no longer needed and is available for use. The data is still there but the location the data occupies may be overwritten when there is a need for the space. Until the data is overwritten, the file still exists on the hard disk and is recoverable. The Recycled folder contained 10 files not yet fully deleted. EnCase® automatically attempts to recovers deleted files with valid directory or MFT records. A search for deleted files and folders using the EnCase® forensic program revealed two folders and 2,247 files which could be recovered. EnCase®, when directed by the user, will search for MFT records in the Unallocated Clusters, then EnCase® resolves the recovered MFT records to data on the volume, and attempts to rebuild the folder structure with children files and folders under parent folders. The recovered folder structure is placed under the virtual Lost Files folder created by EnCase®. The majority of files recovered were html, text, and image files apparently from varying websites. The remainder of the recovered files were Norton Antivirus files. None of the recovered files or the files located in the Recycled folder were relevant to the investigation of intrusions nor pertained to the tk rootkit. The EnCase® screen shot below showed the recovered files. The lost files are only recovered when the EnCase® user chooses to do so and the program searches out MFT records; deleted files are automatically recovered and marked as deleted and/or overwritten.



EnCase Forensic Edition
File Edit View Tools Help
New Open Save Print Add Device Search Refresh
Cases Keywords Search Hits Devices
Table Gallery Timeline Report
$Extend
sabs
DESKTOP
Documents and Settings
etc
Favorites
Inetpub
MOBLAS
Multimedia Files
Program Files
RECYCLER
System Volume Information
temp
WINNT
Lost Files
Text Hex Picture Disk Report Console Filters Queries Lock

	Name	File Created
1843	owssvr[4].htm	11/14/01 08:32:39PM
1844	contents[1].htm	11/14/01 08:29:39PM
1845	search[3].htm	11/14/01 08:28:49PM
1846	art200108[1].htm	11/14/01 08:06:00PM
1847	fblvl1a[1].gif	11/14/01 08:03:16PM
1848	prayer1_copy_small[1].jpg	11/14/01 07:47:56PM
1849	gate2_copy_small[1].jpg	11/14/01 07:47:55PM
1850	gate1_copy_small[1].jpg	11/14/01 07:47:55PM
1851	newsrelease091101b[1].htm	11/14/01 07:47:24PM
1852	lowrynote[1].jpg	11/14/01 07:47:03PM
1853	Post1[1].gif	11/14/01 07:43:55PM
1854	downing[1].jpg	11/14/01 07:43:55PM
1855	_vti_inf[1].htm	11/14/01 07:43:36PM
1856	mj[1].jpg	11/14/01 07:41:33PM
1857	dtreg[16].asp	11/14/01 07:41:33PM
1858	ftcontactbg[1].gif	11/14/01 02:38:40PM
1859	caseinfo[1].htm	11/08/01 02:33:21PM
1860	favicon[1].ico	11/08/01 09:12:15PM
1861	ie_120x090_msn7_smiley[1].gif	11/08/01 09:11:42PM

0069-02-51.117\155.216.51.117\C\Documents and Settings\bonaparte\Local Settings\Temporary Internet Files\Content.IE5\KHQNSXP\bar023c_thm[1].gif

Contents of the virtual Lost Files folder

The red x by the selected file means the lowrynote[1].jpg file was deleted and overwritten with the bar023c_thm[1].gif file. The overwriting file image is displayed, but the MFT entry was still located in the unallocated space.

String Search:

An EnCase® based regular text search included the terms vMz-Bots, ServU, httpodbc.dll, mw.txt, and mc.txt, which were the names of the suspect files installed by the intruder. A search for suspect user names was also conducted and included the terms d00m, tk, Negative, NFE, Admin, and TKbot. A final search was conducted for suspected source IP addresses, which included 213.51.1.133 and 24.148.63.209.

All of the terms noted above were included in an effort to find additional files containing evidence of the intruders. IRC nick names and rootkit files can be discovered in the less than obvious areas of the media (e.g. hidden IRC log files). IP address may be discovered in swap or deleted log files. A search of the suspect files may develop additional sources of where the file came from (e.g. a batch file). The search conducted was only for the keywords listed above in standard text. EnCase® provides the ability for the user to search using global regular expressions post (GREP) and Unicode. With GREP searches the keyword list can be modified to search for variations in keywords or hex or decimal characters. An example would be the GREP expression ##?#?\.##?#?\.##?#?\.##?#?[^#\.], which would find matches for IP addresses with up to 3 digit numbers separated by periods. Unicode searches for the additional character sets used by the Unicode encoding standard (a unique number for every character). A Unicode search would have extend the search time and a GREP search was not necessary for this case, subsequently, the two additional search criteria were not used.

The results of the search hits were summarized briefly below:

vMz-Bots	1 hit	1 file
ServU	4 hits	2 files & unallocated clusters
httpodbc.dll	87 hits	18 files & unallocated clusters
mw.txt	9 hits	1 file & unallocated clusters
mc.txt	11 hits	1 file & unallocated clusters
d00m	20 hits	2 files & unallocated clusters
tk	over 117,000 hits	
Negative	256 hits	Over 50 files & unallocated clusters
NFE	over 13,000 hits	
Admin	over 17,000 hits	
TKbot	3 hits	1 file & unallocated clusters
213.51.1.133	31 hits	1 file
24.148.63.209	1 hit	1 file

The keywords which returned over 13,000 hits were ineffective as the time to review the search hits would be excessive resulting in very little usable results. No additional keywords were discovered for additional searches and no new relevant files were found from the search conducted. The media analysis section of this report detailed the relevant findings.

Conclusion:

Examination of the U.S. Government computer assigned the IP address 123.45.67.89, assigned the FQDN something.army.mil, belonging to the U.S. Army, Fort Dix, NJ, revealed it was compromised by the computer assigned IP 24.148.63.209 on 11:04:53 PM (GMT), 13 October 2002, using a known vulnerability in the Microsoft IIS program. The computer was again compromised using the same vulnerability at 10:00:19 PM (GMT), 31 October 2002, by the computer assigned IP address 213.51.1.133. In both compromises, the subject added additional files to the computer in an apparent means to re-access the computer at a later time. The U.S. Army computer was likely compromised for use of storage space for file sharing as well as usage of the Army's high bandwith capabilities. Upon conclusion of the forensic examination, additional leads which could be followed up on by the case agent were discovered. The additional leads would require coordination with the owner of the computers assigned IP addresses 24.148.63.209 (Figure 5.1) and 213.51.1.133 (Figure 8.1) so as to determine if they were also compromised and/or obtain the users information and any log files. If the owners of the computers using the suspect IP addresses are cooperative, images should be obtained and forensic examinations should be conducted of the computers. IP address 24.148.63.209 was registered to the RCN Corporation, 105 Carnegie Center, Princeton, NJ 08540. The last possible lead obtained from the forensic exam would require coordination with RhinoSoft (Serv-U developers) to determine if they have any information regarding the registry keys discovered (Figures 2.1d and 7.1).

Part III –Legal Issues of Incident Handling

A. Based upon the type of material John PRICE was distributing, what if any, laws have been broken based upon the distribution?

John PRICE is subject to numerous federal laws. The United States Federal Copyright Law 17 § 506 states Criminal Infringement (Subsection a) occurs when

Any person who willfully infringes on a copyright law for commercial advantage or private financial gain or by reproducing, distributing including electronic means during a 180 day period of 1 or more copies of copyrighted works which have a total value of $1000.00 shall be punished as provided in section 2319, 18 USC. Evidence of reproduction or distribution of a copyrighted work, by itself, is not sufficient to establish willful infringement (USHR, 2002, 17-506). The statute also establishes a Forfeiture and Destruction clause in which any person who is convicted of any violation of the statute the court may order the forfeiture and destruction or other disposition of all infringing copies or phonorecords and all implements, devices, or equipment.

If John PRICE were a civilian working for a U.S. Government agency, an additional federal charge of 18 USC § 641, which covers Embezzlement and Theft of public money, property or records could be assessed. The statute reflects if someone embezzles, steals, purloins or knowingly converts for his use or that of another without authority, sells, conveys or disposes of any record, voucher, money, or thing of value of the United States or of any department or agency made under contract for the United States or if someone receives, conceals, or retains the property with intent to convert it to his/her use or gain, when they knew it to be embezzled, stolen, purloined or converted shall be fined or imprisoned not more than ten years, or both, however; if the value of the property does not exceed the sum of $1,000 there is a fine but the prison term will not exceed one year (USHR, 2002, 18-641).

In furtherance of this, if John PRICE was a civilian hired by the Department of the Army (not a soldier), he would also be subject to administrative actions by the Office of Personnel Management for failure to adhere to the Army's Internet Policy located in Army Regulation 25-2, Information Assurance. The policy pertains to both military and civilian employees and specifically addresses copyright protection stating that users shall respect the legal protection provided by copyright, license, and authorship of messages, programs, and data on the network. The copyright laws of the United States provide that the owner of copyright (usually the originator of the work) has exclusive rights to reproduce, distribute, prepare derivative works, and publicly display or perform a work. Unless there is specific notice to the contrary, material on the Internet is protected by copyright even if it does not have a copyright notice (such as the "c" in a circle or the word "copyright" followed by a name and date)" (DA, 1998, Appendix D). The policy generally outlines the laws and regulations a user would be subject to for infringing on copyright laws stating that any user who fails to follow the Army's Internet Policy, these guidelines, or any laws or regulations applicable to Internet use is subject to such action as may legally be taken under the Uniform Code of Military Justice, Office of Personnel Management regulation, or relevant U.S. Code" (DA, 1998, Appendix D).

If PRICE was in the U.S. Army and was to be prosecuted under military law, PRICE would likely be charged with the Uniform Code of Military Justice, Article 92, Failure to Obey Order or Regulation. This article defines the offense as well as the punishment stating that any person subject to this chapter who violates or fails to obey any lawful general order or regulation; having knowledge of any other lawful order issued by any member of the armed forces, which it is his duty to obey, fails to obey the order; or is derelict in the performance of his duties and shall be punished as a court-martial may direct. In violating this Article, PRICE could suffer a Dishonorable discharge, forfeiture of all pay and allowances, and confinement for 2 years for failing to obey a lawful order; a Bad-conduct discharge, forfeiture of all pay and allowances, and confinement for 6 months for disobeying other orders and forfeiture of two-thirds pay per month for 3 months and confinement for 3 months for being derelict in his duties.

B. What would the appropriate steps be to take if you discovered this information on your systems?

If the system was on, attempts would be made to determine if any active connections to file sharing programs or servers was currently in use. A utility such as Log Collector, a forensic incident response tool, would be run to establish those connections as well as capture the unauthorized file names and location on the computer. The computer would be disconnected from the Internet and the power unplugged. The computer would be seized as well as any surrounding media as evidence, to be handled accordingly. Each piece of media and the computer hard drive would be imaged and the image files would also be taken as evidence. The images would be examined for unauthorized copyrighted material, file sharing programs, additional storage locations (e.g. ftp sites, additional disks). Based on the evidence found during the examination an interview of the suspect would be conducted and a confession for his illegal activities obtained. During the course of the entire investigation, actions would be coordinated with any additional law enforcement agencies, the action commander (person responsible for suspect and computer) and the prosecuting attorney. The procedures I would take would follow the Army Regulation, which covers criminal investigations. The regulation provides guidance for all felony investigations as well as computer crime. The steps outlined above were in accordance with CIDR 195-1, Chapter 12 (DA, 2003, Chapter 12). The specific chapter provides guidance on authority to take the steps outlined above as well specific steps to be taken for computer crime related cases. Evidence procedures, and additional agency and attorney coordination are covered elsewhere in CIDR 195-1 and are consistent for all criminal investigations for the U.S. Army.

If operating in a civilian environment, separate but equally important steps would have to be taken to not only isolate the incident and identify the perpetrator(s) but to protect the organization from potential liability. If PRICE, an employee of the company was using organizational resources to steal, hide and subsequently distribute material in violation of copyright laws, he potentially could subject the organization as a whole to civil and criminal action. The system administrator, in concert with the appropriate criminal investigative entity and corporate legal advisor, should conduct an evaluation, with the utmost scrutiny as to PRICE's role in the organization, specifically his level of access to the organizational network. Significant issues which have to be addressed include whether PRICE was granted administrator rights on the company's servers, did he have access to multiple workstations and is it possible he could have stored the stolen material on an organizational asset. Once a determination has been made as to the total access granted to PRICE, a strict review of each device should be conducted to ensure that first, PRICE did not store any stolen material on these device and secondly, that he did not keep his unauthorized program on the devices in an effort to conceal his activities. The corporate legal advisor should closely review the established policies of the organization verifying that PRICE received training and it was documented as to the company policies on the misuse of company equipment. Human Relations personnel should be consulted to determine if PRICE had previous incidents or if he was counseled as to prior violations. The corporate legal advisor should start building a packet documenting PRICE's activities both past and present which could be used by law enforcement and executives in the company for termination action. More importantly, the organization should disclose to the appropriate law enforcement communities PRICE's activities as well as to companies which may have been his victim. The organization should further disclose their actions in regards to correcting not only the violations by PRICE but how they plan on stopping this type of behavior in the future. Lastly, it is important that the system administrator eliminate PRICE's ability to access the organizational network from anywhere. This activity is crucial for the protection of evidence and to avoid future incidents and liability.

C. In the event your corporate counsel decides to not pursue the matter any further at this point, what steps should you take to ensure any evidence you collect can be admissible in proceedings in the future should the situation change?

All felony criminal actions are pursued in a Report of Investigation by the U.S. Army Criminal Investigation Command (USACIDC). Computer crimes are investigated by the U.S. Army Computer Crime Investigative Unit, a unit of the USACIDC. Whether or not the case is prosecuted, the handling and disposition of evidence is always conducted in accordance with Army Regulation 195-5, Evidence Procedures. Any computer equipment received collected is documented on a Department of the Army (DA) Form 4137, Evidence/Property Custody Document and includes all relevant information pertaining to the equipment seized or received. The evidence is the logged into the evidence room and maintained until an authorization for disposition is obtained. If possible, our office collects and image of computer media (e.g. compact, floppy, or hard disks) using imaging utilities such as dd, Safeback, or EnCase, which is also collected on DA Form 4137 and stored in the evidence room. Forensic examinations are conducted on the images obtained in an effort to maintain evidence integrity. Forensic utilities such as EnCase and Autopsy are again used to assist with maintaining evidence integrity by preventing alterations to the media. Network intrusion detection system and computer log files are also collected as best evidence to supplement forensic examinations of compromised computers. As a federal law enforcement agency, we have to be particular of the information received from outside sources, especially system administrators. The information system administrators or Internet service providers give to our office may not be obtained in an illegal manner otherwise the information cannot be used in criminal proceedings. As law enforcement, we cannot direct a system administrator to monitor connections to a compromised computer on his network as he would be considered to be conducting the monitoring under the color of law. The monitoring is subject to approval by the Army General Counsel and cannot be done at the request of USACIDC. The proper handling documentation of all evidence received by our office prevents a loss of evidence in trials. Regardless of the final disposition of a case, our procedures are always followed.

The decision not to pursue additional action against PRICE should not be made by the corporate legal advisor; however, if the organizational leaders leave the decision to their lawyer, certain procedures should be adhered too. The primary effort should not be in preserving evidence; the evidence was already identified. The primary concern should be to determine by what means PRICE was able to complete his criminal act without detection and how the organization as a whole plans to prevent this activity in the future. The data, material, logs and any other evidentiary information should be downloaded and stored, possibly on a CD or DVD. Copies should be provided to the Human Relations Department and the legal counsel and a copy retained by the Systems Administrator in a secured area. Further, the systems administrator should have taken copious notes as to all of his/her activities, documenting the material found, how it was found and where. All of this information could be summoned into a legal proceeding in the future and it is unlikely anyone will remember what happened unless there is adequate documentation.

Based on experience, it is unlikely that the “situation” will change if the company fails to prosecute or pursue PRICE for this infraction. Clearly, violations of US Copyright Laws are a significant criminal issue possibly subjecting PRICE to stiff fines and the organization to civil liability. Most state and federal prosecutors will not entertain the idea of trying to prosecute an offender a year or two years after the offense occurred and will most certainly question the motivation of the corporate legal advisor and the organization as a whole for waiting. Even the best efforts by the organization to “secure” the evidence will most likely not be adequate for legal proceedings. The chain of custody of this critical data will be questioned by defense lawyers with issues such as where was it stored, who had access and how does the systems administrator know the data was not manipulated. Again, this is an organization subjecting itself to extensive ridicule and possible civil actions. The corporate lawyer's best path is to pursue this matter and provide the data to the appropriate law enforcement agency. Provide them with the evidence and full cooperation and then allow the respective prosecutorial agency to make a decision. If there is a declination, the company can still take administrative action against PRICE but more importantly, the company has put the evidence in the hands of the right agencies.

Whether prosecution is pursued or not the organization needs to adjust priorities in protecting their information technology assets through additional training. If PRICE was successful in conducting criminal activities using organizational assets, there may exists an apathetic atmosphere in regards to IT security. Clear and succinct policy letters need to drafted (if they do not exist) explaining to employees the limitations of use of company technology assets. More importantly, these policies need to address what the ramifications are for violations. Clearly, PRICE needs to be terminated for this activity and more than likely prosecuted to the fullest extent of the law. Employees need to know that they will be prosecuted for infractions.

D. How would your actions change if your investigation disclosed that John PRICE was distributing child pornography?

The actions of the investigation would not change dramatically. The investigation would continue to follow Army regulations governing criminal investigations and evidence procedures 195-1 and 195-5. When conducting the forensic examination and analyst would be cognizant of the federal laws regulating the disposition and transfer of the files obtained. The relevant law pertaining to Child Pornography is 18 USC § 2252 and the search conducted would attempt to prove the elements listed in the statute. The U.S. Code noted above is lengthy, but provides all elements of the crime which need to be proven in a forensic examination as well as providing the punishment available for criminal prosecution. If the allegation pertained to a commission of the crime by a U.S. Army soldier, then the articles noted in question A from the Uniform Code of Military Justice would pertain. Military justice is conducted in a court martial environment as opposed to a federal district court, but the general proceedings are somewhat similar and all elements of the offense must be proven by the prosecution as in the federal statute. The statute states that certain activities relating to material involving the sexual exploitation of minors and that any person who knowingly transports in the US or foreign commerce by any means including by computer or mails, any visual depiction involving the use of a minor who is engaging in sexually explicit conduct or if any person knowingly receives, or distributes, any visual depiction that has been mailed or shipped or which contains sexually explicit images of children or reproduces any visual depiction for distribution violates this statute.

The statute reflects that whoever violates, or attempts or conspires to violate 18 USC 2522 shall be fined or imprisoned for not more than 15 years, or both, but if the person has a prior conviction under this statute or any State law relating to aggravated sexual abuse, sexual abuse, or abusive sexual conduct involving a minor or ward, or the production, possession, receipt, mailing, sale, distribution, shipment, or transportation of child pornography, such person shall be fined under this title and imprisoned for not less than 5 years nor more than 30 years (USHR, 2002, 18-2522).

Clearly, the allegation of child pornography changes many of the aspects a system administrator, corporate legal advisor and management would take. Discretion at this level is limited and there would be no choice but to advise law enforcement, specifically the Federal Bureau of Investigation of the potential offenses. It would not be a good idea for the organization not to report this regardless of the potential embarrassment.

The company should be aware that on November 29, 1990, the U.S. Congress enacted 18 U.S.C. 2252 making it a federal crime to possess three or more depictions of child pornography that were mailed or shipped in interstate or foreign commerce or that were produced using materials that were mailed or shipped by any means, including by computer. This statute would be significant in the event PRICE stored any child pornography on a company computer.

The Child Pornography Prevention Act of 1996 amends the definition of child pornography to include that which actually depicts the sexual conduct of real minor children and that which appears to be a depiction of a minor engaging in sexual conduct. Computer, photographic, and photocopy technology is amazingly competent at creating and altering images that have been "morphed" to look like children even though those photographed may have actually been adults. People who alter pornographic images to look like children can now be prosecuted under the law. (NCFEC, 2004)

The Department of Justice Child Obscenity and Exploitation Division provides information regarding The Protection of Children from Sexual Predators Act amending the Victims of Child Abuse Act of 1990 by requiring online service providers to report evidence of child pornography offenses to law enforcement agencies. The Act also amends 18 U.S.C. § 2702(b) of the Electronic Communications Privacy Act of 1986 to create an exception to the general statutory bar against a public provider's voluntary disclosure of customer communications to third parties. The online industry is encouraged to familiarize itself with the Electronic Communications Privacy Act of1996, 18 U.S.C. §2701. The reporting section, 42 U.S.C. § 13032, which is similar to 42 U.S.C. § 13031, Child Abuse Reporting, requires anyone who is engaged in providing an electronic communication service to the public, and obtains knowledge of a violation of the child exploitation statutes, to report such violation to a law enforcement agency or agencies. The Attorney General is in the process of designating a federal law enforcement agency which will be responsible for receiving such reports by electronic service providers. A failure to report is subject to a civil fine of up to $50,000 in the first instance and $100,000 for any subsequent failure. No service provider may be held civilly liable for any action taken in good faith to comply with the reporting requirement (USDOJ, 2002).

If PRICE were found to be distributing child pornography, all additional actions should cease until Law Enforcement can arrive. This would be another example of the need for clear and thorough documentation as to how the images were identified, preserved and located. Additionally, the topics previously mentioned, (i.e. level of access PRICE maintains) would need to be evaluated. The issue of the possession and distribution of child pornography is significant and should be referred to Law Enforcement.

WORKS CITED

Department of the Army (DA). (2003). Army Regulation 195-1, Criminal Investigation Operational Procedures. Washington, DC: Headquarters, Department of the Army

Department of the Army (DA). (1992). Army Regulation 195-5, Evidence Procedures. Washington, DC: Headquarters, Department of the Army

Department of the Army (DA). (1998). Army Regulation 380-19, Information Systems Security, 24-25. Washington, DC: Headquarters, Department of the Army

Manual for Courts-Martial United States (MCM). (2000). IV-23-25.

National Center for Missing and Exploited Children (NCFMEC). (2004). Laws and Regulations. Retrieved January 9, 2004 from http://www.missingkids.com/missingkids/servlet/PageServlet?LanguageCountry=en_US&PageId=1199.

US Department of Justice, Child Exploitation and Obscenity Division (US DOJ). (2002). How to Report Child Pornography. Washington, D.C: Retrieved January 9, 2004 from http://www.usdoj.gov/criminal/ceos/report.htm.

U.S. House of Representatives (USHR). (2002). United States Code. (2000 ed.). Washington, DC: U.S. Government Printing Office. Retrieved October 16, 2003 from http://www4.law.cornell.edu/uscode/.