



Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Advanced Incident Response, Threat Hunting, and Digital Forensics (Forensics
at <http://www.giac.org/registration/gcfa>

System Forensics, Investigations and Response

Exercises in Forensic Analysis

SANS October, 2003 New York
GCFA Practical Assignment v. 1.4 (July 21, 2003)
Beth E. Binde
June 13, 2004

Abstract: This submission for the GCFA practical includes the following sections:

Preliminaries	2
Part 1 - Analysis of an unknown binary.....	3
Part 2 - Perform a Forensic Analysis on a System.....	16
Synopsis of Case Facts	16
System Description	17
Hardware	17
Image Media	18
Media and Timeline Analysis.....	22
Recover Deleted Files	39
String Search	40
Conclusions.....	41
Part 3 – Legal Issues of Incident Handling.....	43
References	45

Preliminaries

The forensic workstation used for these exercises is a Dell Latitude laptop running Red Hat Linux 9.0 with 512 MB RAM and a 30 GB hard drive, a configuration that met the stated requirements for the conference. The forensic workstation has not been connected to any network at any time. When the equipment is required for other uses, the hard drive is physically removed, stowed safely, and another hard drive is swapped in.

The workstation was installed on October 6, 2003 as per the instructions given in preparation for the conference. The media for installation of Linux was downloaded from a known trusted mirror and the media was verified. A copy of VMWARE was installed from a vendor CD and a licensed copy of Windows 2000 was obtained from a known source. Additional forensic tools were installed from CDs provided by SANS. No additional software has been installed on the machine.

The following conventions are present the text:

- Output from commands is in Courier New font and enclosed in a box
- Commands and programs are underlined, for example, strings.

References are handled as endnotes, and the first few letters of the author's name or authoring agency are used to annotate the text. Multiple references from the same author are numbered to distinguish them.

Part 1 - Analysis of an unknown binary

This exercise required download and analysis of a floppy image provided on the SANS web site for the purpose of completion of this certification practical assignment. As advised, the file was handled cautiously--- it was downloaded from the website directly to a new (previously unused) floppy disk. The floppy was then inserted in the floppy drive on the forensic workstation for analysis.

The initial steps focused on exploring the nature of the binary file. The `file` command confirmed that the provided image was indeed a zip archive.

```
binary_v1_4.zip: Zip archive data, at least v2.0 to extract
```

Use of the `zipinfo` command (with the `-l` flag) revealed the further information about the binary image. The output shows that the ZIP file contains a disk image in `dd` format, compressed and a corresponding MD5 checksum of the disk image. The MD5 checksum of the target unknown binary, `prog`, is included as well.

```
Archive:  binary_v1_4.zip   459502 bytes   3 files
-r-----   2.3 unx   474162 bx defN 16-Jul-03 01:03 fl-160703-jp1.dd.gz
-rw-r--r--   2.3 unx     54 tx stor 16-Jul-03 02:14 fl-160703-jp1.dd.gz.md5
-rw-r--r--   2.3 unx     39 tx stor 16-Jul-03 02:14 prog.md5
3 files, 474255 bytes uncompressed, 459030 bytes compressed:  3.2%
```

Therefore, it was immediately evident that the target operating system for the ZIP file was a UNIX variant. The first line provides the name of the archive, the number of bytes in the archive, and the number of files in the archive. Following that is a line for each file in the archive with further details about each file.

The first field provides the file permissions for the file. The format depends upon the operating system that was used to create the archive. In this case, the archive was created under a UNIX file system (see the third field) and so the file permissions reflect a UNIX format-- a leading character to designate a file type, followed by three groups of three characters each, showing read, write and execute permissions for owner, group and other, respectively. All three files are regular files (not directories or named pipes) and all display read permission for the owner of the file. This is the only permission inherent in the first file. The second and third files allow the owner to write (change or modify the file), and users in the same group can read the file. Any other user with access to the system can read the file as well. No one has "execute" permissions on the files.

Moving on to the second field, the version of zip that was used to create the archive is shown. In this case, the zip version number is 2.3. The third field was already mentioned as showing the operating system. The fifth field is two characters, the first of which would be the letter t, indicating that the zip program tagged the file as text, or the letter b, indicating that the file was tagged as a binary. If the file is encrypted, the letters T and B are capitalized. The next character indicates whether there is an extended local header (a lowercase l) or an “extra field” (a lowercase x). An uppercase X indicates that both are present, while a hyphen indicates that neither are. So, the zip program believes that there are two text files and one binary, neither of which are compressed or encrypted. The next field pertains to the compression methodology—there are six possibilities, and possibly a compression sub-method. The two that were used in this archive are stor (no compression) and defN (normal deflating). The last three fields are the modification date and time of the file, followed by the name of the file. The final line gives the compression ratio, a paltry 3% in this example, and the total size of the archive before and after compression. Binary files and images are notorious for low compression ratios, while plain text tends to compress well.

The unzip command was used to extract the files, and subsequently the gunzip command to extract the image file itself. Using the dd command, the f1-160703-jp1.dd image was copied to a fresh floppy. The md5 command confirmed that the files carried the correct checksum; that is, that the generated checksum matched the checksums provided in the files f1-160703-jp1.dd.gz.md5 and prog.md5. They matched, demonstrating that the integrity of the files was preserved in the transfers described above. The examination of the floppy itself followed.

The decision was made to use the Autopsy Forensic Browser [CAR2]. One reason for using it was that the SANS System Forensics, Investigations, and Response training was taught using that set of tools. Not only is Autopsy highly recommended, it is familiar, and it is free, this last criteria being nearly always crucial to security departments in the public sector. The structure of the program tends to lead the investigator through the basic steps of the investigation in an orderly fashion. It is to be further noted that Autopsy has provisions for time stamping of activity and making notes internal to the program.

The initial effort used tools such as fsstat to gather general information about the image. The output confirmed the access dates of the files and the operating system as guessed by zipinfo.

```
=====
FILE SYSTEM INFORMATION
```

```
-----
File System Type: EXT2FS
Volume Name:
Last Mount: Wed Jul 16 02:12:33 2003
Last Write: Wed Jul 16 02:12:58 2003
Last Check: Mon Jul 14 10:08:08 2003
Unmounted properly
Last mounted on:
Operating System: Linux
Dynamic Structure
InCompat Features: Filetype,
Read Only Compat Features: Sparse Super,
```

```
META-DATA INFORMATION
```

```
-----
Inode Range: 1 - 184
Root Directory: 2
```

```
CONTENT-DATA INFORMATION
```

```
-----
Fragment Range: 0 - 1439
Block Size: 1024
Fragment Size: 1024
```

```
BLOCK GROUP INFORMATION
```

```
-----
Number of Block Groups: 1
Inodes per group: 184
Blocks per group: 8192
Fragments per group: 8192
```

```
Group: 0:
```

```
  Inode Range: 1 - 184
  Block Range: 1 - 1439
    Super Block: 1 - 1
    Group Descriptor Table: 2 - 2
    Data bitmap: 3 - 3
    Inode bitmap: 4 - 4
    Inode Table: 5 - 27
    Data Blocks: 28 - 1439
```

Since Mr. Price was accused of and suspended for distributing copyrighted material without permission, a Google search was undertaken to locate the GIF files found on the floppy drive. The file [sect-num.gif](#) was immediately located at the Computer Forensics International web site, along with a resized and renamed [sectors.gif](#), named [eviden3.gif](#) on the web site. The images can be viewed at the following web page. www.cf-intl.com/evidence_recovery_basics.htm. A copyright notice is prominently displayed at the footer of the page where this images were found. The HTML documents, [Sound-HOWTO.html.tar](#), [MP3-HOWTO.html.tar](#), [DVD-Playing-HOWTO.html.tar](#) each carry a copyright notice within them. Presumably, the auditors

have ascertained that this material (or other copyrighted material) is being used outside of the copyright agreement. A copyright attorney could advise as to whether the presence of the material on a floppy disk is a violation in and of itself; that may fall under "fair use" of the material.

After the initial examination of the floppy disk, identifying the prog binary was the focus. The output of the strings command seemed to provide a number of possible leads. The best (most interesting) of them follow:

```
Keld Simonsen
C/o Keld Simonsen, Skt. Jorgens Alle 8, DK-1615
Kobenhavn V
Wrong medium type
No medium found
Disk quota exceeded
Remote I/O error
Is a named type file
Read-only file system
Illegal seek
No space left on device
File too large
Text file busy
```

A search on the name "Keld Simonsen" yielded information about a person much interested in internationalization standards in programming. There is a multitude of references to the work of Keld Simonson available through a Google search, so it seems to be happenstance that his name turned up, and so this was tagged as a flase lead. The other phrases were suggestive, however, that this may be some sort of file manipulation program. The program was carefully invoked on a test Windows system, not connected to the Internet, using only the -help option which yielded the following:

```
prog:1.0.20 (07/15/03) newt
Usage: prog [OPTION]... [<target-filename>]
use block-list knowledge to perform special operations on files
```

If the information is accurate rather than a deliberate decoy, it would tend to confirm the guess about a file manipulation program. The use of "newt" is also suggestive of a possible author or group of authors for the program.

A search on the phrase "use block-list" brought up a reference to the program bmap on <http://old.lwn.net/2000/0420/announce.php3>. An announcement about the bmap program was posted there, along with a link to the Freshmeat site which unfortunately is now defunct. However, now armed with a program name, a subsequent search led to the "Linux Data Hiding and Recovery" article by Anton Chuvakin (CHU).

The article contained a pointer to the source for bmap as well. The source can be found and downloaded here [RID]: http://ftp.scyld.com/pub/forensic_computing/bmap/

In the next screen shot, the file/MAC time information (last modified, last accessed and last changed) are shown. As the local time zone for the incident is unknown, the time zone here was assumed for the case. The last modified time is July 14, 2003 at 9:24 AM. The last access time is July 16, 2003 at 1:12 AM and the last changed time is July 16, 2003 at 1:05 AM. Either this individual works at night, or is in quite a different time zone than this one. The file owner has a UID of 502 as well as a GID of 502. Without the /etc/passwd file to make the correlations, the corresponding username and group name remains unknown, although it may be available from a system backup. The file size of prog is 487,476 bytes.

The screenshot shows the Autopsy forensic tool interface. The top menu bar includes File, Edit, View, Favorites, Tools, and Help. The address bar shows the path A:\autopsy.html. The main window has tabs for FILE ANALYSIS, KEYWORD SEARCH, FILE TYPE, IMAGE DETAILS, META DATA, DATA UNIT, HELP, and CLOSE. The FILE ANALYSIS tab is active, displaying a directory listing for /mnt/floppy/. The listing includes columns for DEL, Type, NAME, MODIFIED, ACCESSED, CHANGED, SIZE, UID, GID, and META. The file 'prog' is highlighted, showing a size of 487,476 bytes and a UID/GID of 502/502. The 'Generate MD5 List of Files' button is visible above the table.

DEL	Type	NAME	MODIFIED	ACCESSED	CHANGED	SIZE	UID	GID	META
	d/d	../	2003.07.16 01:03:13 (EST)	2003.07.16 01:12:39 (EST)	2003.07.16 01:03:13 (EST)	1024	0	0	2
	d/d	./	2003.07.16 01:03:13 (EST)	2003.07.16 01:12:39 (EST)	2003.07.16 01:03:13 (EST)	1024	0	0	2
	r/r	..5456g.tmp	2003.07.14 09:13:52 (EST)	2003.07.16 01:11:36 (EST)	2003.07.14 09:13:52 (EST)	2592	0	0	28
	d/d	Docs/	2003.07.14 09:22:36 (EST)	2003.07.16 01:10:01 (EST)	2003.07.14 09:43:44 (EST)	1024	502	502	15
	d/d	John/	2003.02.03 06:08:00 (EST)	2003.07.16 01:09:35 (EST)	2003.07.14 09:49:25 (EST)	1024	502	502	12
	d/d	lost+found/	2003.07.14 09:08:09 (EST)	2003.07.16 01:06:15 (EST)	2003.07.14 09:08:09 (EST)	12288	0	0	11
	d/d	May03/	2003.05.03 05:10:00 (EST)	2003.07.16 01:09:49 (EST)	2003.07.14 09:50:15 (EST)	1024	502	502	14
	r/r	nc-1.10- 16.i386.rpm..rpm	2003.07.14 09:12:02 (EST)	2003.07.14 09:12:02 (EST)	2003.07.14 09:43:57 (EST)	56950	502	502	22
	r/r	prog	2003.07.14 09:24:00 (EST)	2003.07.16 01:12:45 (EST)	2003.07.16 01:05:33 (EST)	487476	502	502	18

The MD5 checksums for the file are taken from the autopsy program, which renders them as text files when the "Generate MD5 List of Files" option is chosen above.


```
MD5 Values for files in /mnt/floppy/ (images/fl-160703-jp1.dd)
```

```
7b80d9aff486c6aa6aa3efa63cc56880 prog  
535003964e861aad97ed28b56fe67720 nc-1.10-16.i386.rpm..rpm  
f13ddc8775e4234f8d889a6e49bc69eb .~5456g.tmp
```

According to Chuvakin, the bmap program is used for hiding data in slack space on Linux systems. Typically the number of blocks allocated for a file is larger than what is actually needed to store the data, and some disk space is wasted. The Linux file system, ext2, uses addressable parts of the disk called blocks, which have the same size across the disk geometry. Block sizes of 1, 2 or 4 KB are generally typical of an ext2 file system. It follows that if a file is smaller than the block size, or if the last portion of the file does not fill up an entire block, there will be unused space in the remainder of the block (or blocks) allocated to the file. This unused space is called “slack space”.

The bmap program can insert data into slack space and even wipe it out again, if necessary. It can query for available slack space as well as place data there. Further, the data is invisible to the file system, undetectable by file integrity checkers (check sum programs) or disk usage programs. For smaller block sizes, smaller chunks of data can be hidden. The inverse is true as well, that large block size will allow for larger chunks of data to be hidden. Forensic tools can discover and recover the data, but it is time consuming and requires considerable expertise. The bmap program is quite useful for storing secrets and has possibilities for planting evidence as well [CHU].

With regard to forensic footprints, the user of the binary seems to have taken care to limit them as much as possible. The ldd command shows that the binary is statically linked, not dynamically linked, so no system libraries are loaded when the program is executed. They have been placed directly in the executable itself.

In order to discover more about the footprint of the program, the strace program was used in an attempt to analyze it. The example used is taken from the Chuvakin article cited earlier. [CHU] and preceded by the strace command in order to show what system calls are made by the prog executable. The command line used was:

```
echo "evil data is here" | strace /tmp/prog -p /tmp/md5
```

The /tmp/prog file is the binary under investigation, and it was used to stuff the string “evil data is here” in the junk file /tmp/md5. To give a synopsis of the activity, the program opens the target file, determines the size of the file to be 489 bytes and calculates the slack space as 3607 bytes. It then seeks to the end of the slack space, writes the specified data following byte 489, and exits. The output from strace is presented here in its entirety, as it is short enough to display it all:

```

[root@LinuxForensics tmp]# echo "evil data is here" | strace /tmp/prog --p
/tmp/md5
execve("/tmp/prog", ["/tmp/prog", "--p", "/tmp/md5"], [/ 34 vars *]) = 0
fcntl64(0, F_GETFD) = 0
fcntl64(1, F_GETFD) = 0
fcntl64(2, F_GETFD) = 0
uname({sys="Linux", node="LinuxForensics", ...}) = 0
geteuid32() = 0
getuid32() = 0
getegid32() = 0
getgid32() = 0
brk(0) = 0x80bedec
brk(0x80bee0c) = 0x80bee0c
brk(0x80bf000) = 0x80bf000
brk(0x80c0000) = 0x80c0000
lstat64("/tmp/md5", {st_mode=S_IFREG|0644, st_size=489, ...}) = 0
open("/tmp/md5", O_RDONLY|O_LARGEFILE) = 3
ioctl(3, FIGETBSZ, 0xbfffecb4) = 0
lstat64("/tmp/md5", {st_mode=S_IFREG|0644, st_size=489, ...}) = 0
lstat64("/dev/hda2", {st_mode=S_IFBLK|0660, st_rdev=makedev(3, 2), ...}) = 0
open("/dev/hda2", O_WRONLY|O_LARGEFILE) = 4
ioctl(3, FIGETBSZ, 0xbfffec24) = 0
brk(0x80c2000) = 0x80c2000
ioctl(3, FIBMAP, 0xbfffecb4) = 0
write(2, "stuffing block 886258\n", 22stuffing block 886258
) = 22
write(2, "file size was: 489\n", 19file size was: 489
) = 19
write(2, "slack size: 3607\n", 17slack size: 3607
) = 17
write(2, "block size: 4096\n", 17block size: 4096
) = 17
_llseek(4, 3630113257, [3630113257], SEEK_SET) = 0
read(0, "evil data is here\n", 3607) = 18
write(4, "evil data is here\n", 18) = 18
close(3) = 0
close(4) = 0
_exit(0) = ?

```

While it is interesting to see exactly how the process works and what system calls are made, it added only marginally to the understanding of the program in this instance.

To compare versions, the bmap rpm was installed on a Red Hat Linux 9 system and the binary was statically compiled on the same system, using the simple expedient of mimicking the commands generated by the makefile supplied with the source, and modifying the cc commands to include the -static switch to force static libraries to be used (rather than loading the various libraries dynamically when the program is executed). The size of the statically compiled bmap binary weighs in at 652,906 bytes. The rpm version of the file (with dynamically loaded libraries) is 116,868 bytes, a ratio of approximately 6 to 1 in terms of size difference. The version of bmap found on the recovered floppy is in the middle at 487,476 bytes. Since it is statically compiled, it would be expected that the size would be significantly larger. However, since it is not

the same size as the rpm version of the file, it seems as if there has been some customization on the recovered version. Looking at the output from the -help switch shows that the flags for the modes have been compacted to a single letter, such as “p” for “putdata”, but there may be other customizations as well that would require binary analysis or a disassembler such as Ollydbg. [SKO]

For example, the email address of the author appears as newt@scyld.com in the version that was statically compiled on the forensic workstation. The full email address also appeared in the RPM version that was downloaded to another Red Hat Linux 9 host. However, it is truncated to "newt" on the recovered version, and the bmap program is renamed to prog on the floppy. Partial output from the strings command follows:

```
1.0.20 (07/15/03)
newt
use block-list knowledge to perform special operations on files
prog
main
```

The date, in this compilation of bmap, is 07/15/03, and represents the date that the binary was compiled. This held true for the compile done on the forensics workstation as well as for the compile that is part of the rpm available at the ftp site for the program, maintained by the person who originally wrote and still maintains the program. However, this is contradicted by the create date given by autopsy of 7/14/2003. A logical explanation is that this is circumstantial evidence indicating that the program may have been compiled on another machine and then written to a floppy. Another possibility is that the file creation dates were altered. An example of a commercial program that can do this is filetweak, found on the web at this location: <http://www.febooti.com/products/filetweak/>. AttributeMagic offers a similar utility with a 30 day free trial download from http://www.attributemagic.com/attributemagic_pro.html. Doubtless there are freeware versions as well for backdating files. Although it is possible, evidence to support this conjecture, has not yet been located, so it is purely speculative. However, this may provide a possible avenue of questioning in an interview with the subject.

As the floppy image is examined further, it is also noted that a deleted copy of prog seems to have been resident on the floppy as well, although there is no information remaining other than that it existed—the modified, accessed and changed times are all zeroed out, as well as the size, uid, gid and meta data. Since other deleted files are not zeroed out, this might support an attribute modification theory.

There is other evidence of customization as well. The rpm for netcat on the disk is different from the standard rpm file, first due to the "double" file extension of “rpm..rpm” instead of simply “rpm”. A careful installation of the rpm was done to a copy of the a floppy.

```
rpm --install --relocate /usr=/mnt/floppy nc-1.10-16.i386.rpm..rpm
```

The resulting binary contained dynamically loaded libraries and gave its version number as 1.10. The known clean version of netcat on the forensic laptop had static libraries, the same version number, but a radically different file size of 439,240 (while the floppy version size was 22,199 bytes). Although the strings output of bmap is similar, the man page is the same and the internal description seems consistent, the evidence leads to the conclusion that the source was individually tailored to change the author information and internal compilation date. Other changes of an unknown nature were required to substantially impact the size of the binary file. Since the program is licensed under the GNU General Public License, this does not violate any copyright agreement. The file COPYING in the source tar ball for bmap explicitly declares that program modifications are permitted, as per a typical GNU General Public License.

The remainder of the floppy was examined for other evidence. One of the findings is a tar archive of html files. The html files are DVD-Playing-HOWTO-html.tar, Kernel-HOWTO-html.tar.gz, MP3-HOWTO-html.tar.gz, and Sound-HOWTO-html.tar.gz. There is also a deleted version of the DVD-Playing-HOWTO-html.tar file with zero values for modified, accessed and changed times, as well as size, uid, gid and meta data. An examination of this area under autopsy showed that it was a Microsoft executable was located at fragment 165. The file type was flagged as an 8086 relocatable (Microsoft). Extensive efforts to isolate the file for further examination were fruitless but should be pursued further. This is inside the Sound-HOWTO-html.tar.gz compressed archive. When the carve option of bmap is used to try to extract data, a fragment appears near the beginning of the archive (in the Sound-HOWTO-1.html file) that includes the nuucp command (an outdated early file transfer protocol). At fragment 281 (inside the prog program), a Dbase 3 data file with memo(s) was found, in at fragment 327, an ms-windows icon resource was located, and at fragment 341, another 8086 relocatable. More interesting is a SysEx file, associated with MIDI (sound) output at fragment 350 inside the sec-num.gif file and within the same file, a Dbase 3 data file at fragment 357. In the file sector.gif in the directory John, fragment 379 contains a file type "Sendmail frozen configuration". Fragment 380 (in the same file) claims to have a CLIPPER COFF executable (used on a VAX system) and a Dbase 3 data file at fragment 385. At fragment 414, the Hitachi SH big-endian COFF object was found. However, this increasingly odd collection of file types began to arouse suspicions.

Using the instructions from Chuvakin as a guide, the attempt was made to extract the possibly hidden files using the command

```
bmap -mode slack <filename>.
```

The usage was tested using the example given in the text, then a GIF file was stuffed and recovered. Attempts were made to recover hidden files using bmap on almost all of the files on the floppy image before giving up, finally using the prog program itself on the sector.gif file since it advertised the most intriguing possibilities. The most facile

conclusion is that the data fragments happened to have data that matched the “magic” number for the possibly inserted files. [KES], [PAR]. There is the tantalizing possibility that the prog version of bmap has an undocumented feature that permits data to be hidden and recovered differently, even though the program, when tested, behaved similarly and allowed recovery of data—the structure of the command line was slightly altered but the program appeared to behave the same way in testing.

One embedded 8086 relocatable file was exported to a floppy and examined on a Windows test host, following a standard examination paradigm (SKO). The strings Command showed no readable output. When executed under the Ollydbg analyzing ger [YUS], the following libraries were displayed: RPCPT4.dll, user32.dll, ntdll.dll, adrapi32.dll, and kernel32.dll. It appears that these are standard libraries that are automatically loaded upon execution of a Windows program. [MIC1] The same libraries were loaded when a completely different executable was loaded into the debugging tool. Machine code appeared in the debugger, but seemed to be loaded in from the various DLL files. When the executable was cautiously executed, it simply exited. At this juncture, the conclusion is that random strings of bytes that correspond to “magic numbers” appeared in the files. Extended efforts to uncover further information about these files were not productive.

As the workstation was wiped before it could be examined, it will be difficult to prove that the bmap program was executed on the workstation. Even though the floppy was found in the disk drive, Mr. Price denies that it is his. There is unlikely to be logs that show which commands were executed as syslog cannot be used to store such information on a central server. Network logs or other logs sent to a central syslog server would help confirm the time that the system was wiped, since log activity would dramatically drop when the system was completely wiped. Additional circumstantial evidence should be sought in terms of Mr. Price's arrival time and departure time for that work day. Video surveillance logs or employee sign in may place Mr. Price on the scene at the crucial time.

Institutions (other than Universities, that is) tend to have policies restricting the installation of software. That issue and possibly other violations of company policies may be avenues to pursue strictly in terms of employee disciplinary action. Depending on the particular job function of Mr. Price, wiping out the workstation is likely to have destroyed company data. One hopes that a robust backup system was in place already, but it assuredly will be put in place following these events, if it is not already. It is further anticipated that stricter controls and tighter audits will be imposed as well.

Without further information about the institution where this event took place, it is difficult to estimate the financial (or other) impact on the company. At the very least, the systems administration staff to re-install the machine, possibly at significant cost in time (and therefore dollars). This may prove to be a demonstrable loss, although probably not a substantial one in view of the general obligation of the company to provide for business continuity. This might be required if the institution is subject to the provisions of Gramm-Leach-Bliley Act of 1999 [US], also known as GLBA.

Financial institutions are subject to the provisions of GLBA. The legislation provides for increased regulation, especially with regard to customer privacy. This would make the improper disclosure of customer financial information a violation of the law. With regard to Universities, the privacy of student data is already covered by other legislation such as the Family Education Rights and Privacy Act (FERPA). However, Universities are subject to the FTC Safeguarding Provisions, requiring that privacy of student financial data be preserved. [NAC]. The FTC requires that financial information be secured, and that if it is in electronic format, the computers be secured as well. The transmission of data is required to be secured. The FTC further requires that data be disposed of securely, by shredding paper and wiping disk drives. There are further provisions for managing system failures, requiring development of a business continuity plan. Patching and software updates are strongly recommended along with anti-virus software that updates automatically. [FTC]. Therefore, if it was shown that Mr. Price was disseminating financial data or had irretrievably destroyed financial data, the company (or University) would be in trouble with federal authorities. While it was first thought that GLBA applied only to financial institutions, subsequent administrative rulings have extended the reach to educational institutions. The thinking is that since large universities collect monies, make loans, and store sensitive information about students and the financial affairs of their parents, the responsibility to maintain the privacy and confidentiality of the data is similar to that of traditional financial institutions.

We now move back to evaluating the damage that has certifiably been done by the employee with the wiping out of the work station. Under the New Jersey statutes, which apply in the state where I live in the United States of America, the offense is most likely to be classified as a third degree offense (damages more than \$500 but less than \$75,000) [NJ1]. While keeping in mind that this paper does not intend to provide legal advice, it seems as if the specific offense is covered by the statute found on the following web page:

2A:38A-3. Computer
related offenses; compensatory and punitive damages; costs and expenses

and cited as "the purposeful or knowing and unauthorized altering, damaging, taking or destroying of a computer, computer system or computer network". The copyright violations would fall under the Digital Millennium Copyright Act (DMCA). Since the auditors have tagged Mr. Price for violations under the DMCA, it will be presumed that the use of the files did not fall under the educational exemptions in the law.

After preliminary investigation, the interview with Mr. Price should have at least two people present, one to ask questions and the other to observe and take notes. The usual rule of thumb is to ask only questions for which the answers are already known. The alleged perpetrator may not have an interest in answering any questions at all, and may or may not be truthful. It would also be wise for him to request a union representative or attorney be present during questioning. Having said all of the above, the list of questions below do not reflect what would likely be asked, given

that little is known about the surrounding circumstances or the particular role we would be assuming in this scenario. Instead, it reflects what we would want to know from Mr. Price if permitted to speak with him and with a reasonable expectation of truthfulness.

So, the initial questions designed to put Mr. Price at ease, establish his identity and verify his position in the company will be assumed, along with questions about whether or not he was on the premises.

- Did you compile and install the prog program?
- Does anyone else have access to your workstation?
- How (and why) was the workstation was wiped out?
- The audit department reported that copyrighted material was being distributed (describe some of the supporting evidence). Do you have any information about it? Do you have an explanation as to why they may have drawn that conclusion?
- What explanation do you have for this note?

Hey Mike,

I received the latest batch of files last night and I'm ready to rock-n-roll (ha-ha).

I have some advance orders for the next run. Call me soon.

JP

Depending on the information provided by Mr. Price, the investigators will have follow up questions and further avenues to explore.

To summarize the recommendations to the systems administrators on site:

- Preserve the floppy disk.
- Preserve any log information about the workstation. Possible sources are a syslogs sent to a central server, ftp server logs, firewall or IDS logs, netflow logs. Be sure to set aside any backups containing logs to prevent them from being overwritten. This will help support prosecution for the wiping of the workstation and copyright violations.
- Follow the corporate policy with regard to contacting law enforcement. Whether there is already a policy in place or not, this decision will involve senior management and corporate counsel.
- If the decision is made not to contact law enforcement, search the office for other media that may contain relevant information. Be sure to include zip drives, CDs, other computers and electronic devices in the office, and paper records. Keep a careful written record of any items removed from the office and preserve those items carefully.
- Review the evidence collected by the audit department with regard to the copyright violations. Reportedly, there was enough evidence to suspend Mr.

Price from his place of employment. This evidence may cast some further light on the events. The material is copyrighted, and there may be violations of the copyright law.

- Preserve the workstation in its current state until the matter is entirely resolved. Depending on the nature of the data that was being passed, the Department of Defense Computer Forensics Laboratory [DCFL] or private forensic organization might be called in to recover data from the disk drive.

A number of references used as background for this paper are cited at the end of the paper. Crane [CRA], Chuvakin [CHU] and Ridge [RID] (the author of bmap), and Ed Skoudis [SKO] are among the most useful for further reading.

© SANS Institute 2004, Author retains full rights

Part 2 - Perform a Forensic Analysis on a System

"Electronic evidence is, by its very nature, fragile." [NIJ, p. 2]

The target system of this forensic analysis held data of a sensitive nature. Therefore, details that would identify the department or anything about the nature of the data have been redacted from the following account.

Synopsis of Case Facts

The compromise was initially reported at 10 AM on Monday morning, April 26, 2004. A telephone call came in from a department head at almost the same time as an email message was received from the system administrator, notifying the central security office of the intrusion. On Monday morning, the system administrator went to the server room to investigate a non-responsive system. Noting the infamous "blue screen" on the Windows server, the first assumption was that the machine had crashed. However, after rebooting, an icon for a remote administration program was noted. As this program was not installed by the system administrator, this was the first indication of a compromise.

The system administrator examined the services running on the machine and found a remote administration program. This was renamed, along with the other programs left by the hacker. A rogue File Transfer Protocol (FTP) program was found listening on the system, and the RemoteAdmin icon was noted in the system tray. The sysadmin changed the file names in a pattern, appending the .exe extension to the original file name and substituting .bak for the extension, so that `info.exe` became `infoexe.bak`. The creation date on these files was around 7:50 PM on April 23. The target host was then removed from the network, although it was left up.

The primary concern of the department was to determine whether or not sensitive data had been accessed during the time that the intruders were active on the system. The local system administrator, eager to help, wished to examine the system further for information about the hacker activities. The first task was to explain that any access to the system would subject the image to possible damage. Next, the system administrator was advised to configure a new system as quickly as possible to free the original for a forensic examination. The department head agreed, and as soon as the configuration of the replacement system was completed, it was shut down normally.

The generally recommended procedure is to remove power from the computer, and then move the system. [NIJ, page 33]. However, since the system had already been rebooted, it was decided to execute a normal shutdown. The monitor, keyboard and

mouse were removed. The tower was wheeled into the central security office. Ticket number 41994 was assigned to the incident.

System Description

The installed operating system was Windows 2000 server running IIS and Oracle It received FTP feeds of data from another system and this data was made available to clients via FTP. The system acted a web server providing crucial services and information to the public. Finally, the system was also a departmental email server.

In general, patches were kept up to date on the system, except for the most recent set announced by Microsoft on April 13, 2004 [MIC3]. The system administrator asserted that these were installed on Monday morning, April 26, after an lsass error was noted in the event logs. Following the installation of patches and changing system passwords, a call was placed to the central security office.

Hardware

The target system is a Gateway 7210 server with a ship date of May 2, 2000. The tower was tagged with the ticket number 41994. The system serial number was noted, but the decision was made to suppress the serial number from this report. Information about the hardware configuration is available to anyone who has the serial number. Although this seems innocuous enough, it was decided to redact identifying information.

The tower case is 28.8 inches high, 8.6 inches wide, and 17.4 inches high. There is an internal CD-ROM drive as well as a floppy drive. The installed memory is 256 MB. The configuration includes 3 Western Digital hard drives, 10,000 RPM and 9.15 GB each. There is a single disk controller, an EDO AdacSCSI Disk Array Controller, which supports RAID. The average seek time for the drives is listed as 5.2 ms for a read and 6.2 ms for a write operation. The disks are configured with RAID5 as two file systems, designated as C: and D: The operating system is installed on the C: drive and data is kept on the D: drive. The size of the C:\ file system is about 2 GB and the D:\ file system is about 15 GB. The disk drives were not removed in order to make a forensic copy, so the serial numbers were not noted. [GAT]

The L2/L3 mappings were searched for confirmation of the MAC address. Briefly, these mappings associate IP addresses with Ethernet addresses and also provide the first time and last time that the host was seen on the network. The references to L2 and L2 come from the Open Systems Interconnect (OSI) model describing network communications. Layer 2 (L2) refers to the data link layer, which defines the access strategy for sharing the physical medium; access is governed by the Media Access Control (MAC) address for each device in an Ethernet LAN. Layer 3 (L3) to the network layer, which relates to connectivity over different network segments. The L2/L3 mappings show no entries for this host after mid-January, 2004, although obviously the host has been up and working. It was confirmed with the system administrator that this was the date of the installation of a Cisco PIX firewall. [OSI]

In this installation, the firewall handles traffic for the Local Area Network (LAN) behind it, and the firewall routes the traffic for the LAN behind it. The public network addresses that had previously been assigned to the hosts on the LAN were preserved. A frequently used alternative addressing scheme would assign private, non-routable IP addresses to the hosts behind the LAN and perform Network Address Translation. However, this would also require re-numbering the host IP addresses of every host on the LAN simultaneously with the installation of the firewall. The choice was made to stay with the public IP addresses. The firewall was set up in “pass through” mode in January. No ports or IP addresses were blocked. At the time of the intrusion, no additional rules had been set.

The first 6 characters of the Media Access Control (MAC) address is uniquely assigned by the Institute of Electrical and Electronics Engineers to manufacturers of network interface cards. This virtually guarantees that a MAC address will be unique within a LAN segment. The manufacturer of this particular card is listed as the NKE Corporation, 27 Zusho Baba, Nagaokakyo-shi Kyoto 617-0828, Japan. It is unknown as to whether the Gateway Computer Corporation installed network cards manufactured by NKE or whether this is a replacement network card.

Image Media

The file systems were configured are using RAID, which stands for Redundant Array of Independent Disks. The purpose of RAID is to provide more reliable disk storage and larger virtual disk sizes than what would be available with single hard disks. Although RAID can be implemented in either hardware or software, this is an instance of a hardware implementation. A hardware implementation of RAID requires (at a minimum) a special-purpose RAID controller to handle the management of the disks. Instead of writing files sequentially on the disk, RAID optimizes access by striping. This means that the data is broken down into blocks, and successive blocks are written to separate disk drives instead of sequentially in available free space on a single disk. For each data block that is written, a parity block is generated within the same stripe. The disk used for the parity block is staggered from one stripe to the next. The parity blocks are read when a read of a data sector results in a CRC error. That is, a failure of the Cyclical Redundancy Check (CRC) which is used to guarantee that there has not been a failure or error in the transmission of data. At that time, they are used to reconstruct the data. This hides the data error from the computer (and the user). If an entire disk fails in the array, the parity blocks from the remaining arrays are combined with the data blocks from the remaining disks to reconstruct the data. This data recovery process permits a disk that has failed can be removed completely. The process is referred to as “hot swapping”. However, if another disk is lost, the data is no longer recoverable. A minimum of 3 drives are required in order to implement RAID level 5. [ENC]

Because of the way that data is spread across physical disks, the dd program would not be useful for imaging the disks. The result would be a hodgepodge of various pieces of files without any means of linking them together. In order to produce images, the

system was booted from the Red Hat Linux 9.0 distribution media, exiting at the single user prompt instead of proceeding with an operating system installation. Because the RAID implementation was in hardware, the two file systems were recognized and mountable after Linux probed for all available devices.

Instructions for running the `dd` program were followed from an online article by Brian Carrier [CAR1]. He provides information on how to use the `dd` program under Linux, as well as under Windows. First a USB destination device was mounted:

```
mount -t reiserfs /dev/sdb1 /mnt/usb
```

then the copy was performed:

```
dd if=/dev/sda1 of=/mnt/usb/dd.img.p1
```

```
dd if=/dev/sda2 of=/mnt/usb/dd.img.p2
```

and MD5 check sums were taken. The images were initially copied to a BUSLINK USB2.0 Hard Drive. Both the original tower and this USB disk were retained in a locked office with limited access while the forensic examination was underway.

The disk was mounted as a reiserfs file system in order to perform the copy. This file system, originally designed by Hans Reiser, reduces the chance of file system corruption. This is due to the implementation of file system journaling, which means that the file system keeps a log or record of changes to the data areas of the disk. If a system crash occurs, anything that was lost during the crash can be re-created. Additionally, reiserfs handles small files in a more optimal fashion by using a tree data structure to locate and retrieve files. This same tree structure is also used to keep exact byte counts of files. Less space is used than in systems where entire blocks of data must be allocated to files, regardless of the actual size of the files. [COU]

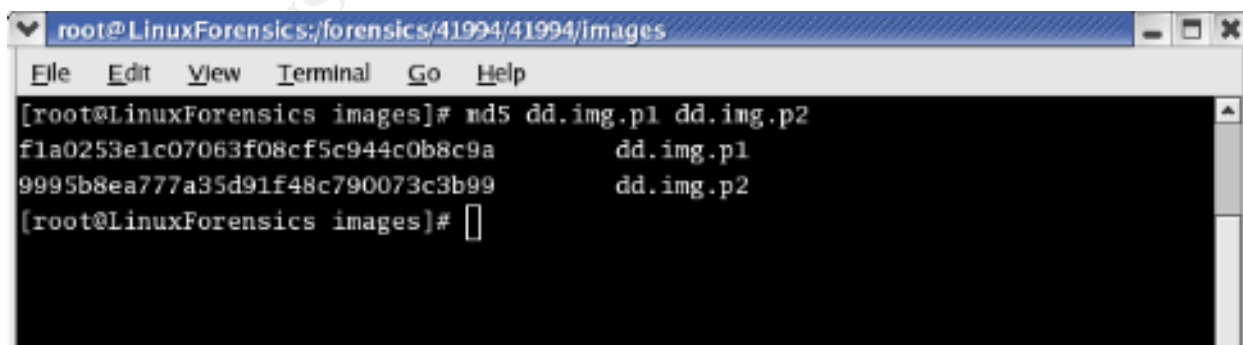
Subsequent to using reiserfs to copy the file systems, further research called the use of reiserfs into possible question. On June 12, a discussion on the forensics-digest listserv hosted by SecurityFocus noted the work of a forensic examiner who unearthed this issue as part of a validation study on forensic tools. (This message is not yet available on the mailing list archive, generally located at <http://www.securityfocus.com/archive>). The validation study was on the use of Knoppix as a forensic tool and compared the MD5 checksums of forensic images created under several different conditions. One set of conditions that was tested involved the use of reiserfs mounted under a Knoppix CD. It was observed that the MD5 check sum did not match the MD5 check sum generated by other disk imaging techniques, which all matched each other. [BAC] Thinking about this further, the journaling provided by reiserfs most likely accounts for the difference in the images. While the integrity of the data copy is sound, it would be possible, by reproducing Mr. Baca's tests, to call the integrity of the copy into question. However, another post by Mr. Nathan Catlow that is already in the archive area, (to be found here: <http://www.securityfocus.com/archive/104/364649>) asserts that the burden of proof

would be on the opposing counsel to demonstrate that the use of reiserfs actually introduced new documents or pictures of an incriminating nature to the disk image. Mr. Catlow believes that reiserfs is a valid tool for securing the disk image, and that it cannot be shown that significant changes are made to the data. Since Mr. Baca performed his testing in May, 2003 and the issue is still discussed, it would appear that this is a technical issue on which strong opposing opinions will remain indefinitely. However, it bears further research in terms of developing procedures for quality investigations. The decision to use reiserfs or not to use reiserfs must be made with a full awareness of the implications.

It took about 12 hours for both disk image copies to complete. They were left to run overnight. The MD5 check sums were also run overnight as significant additional time was required to complete it all. The assignment requirement was to perform an MD5 checksum on the images to verify that they had not changed after recovery from the original system. The MD5 (Message-Digest Algorithm 5) is commonly used for this purpose. Although it was shown in 1994 that special pairs of messages can have the same hash, it is far superior than Cyclic Redundancy Check (CRC) in terms of testing for data integrity. CRC cannot, in fact, guarantee data integrity. As mentioned above, it is intended to detect errors in transmission and duplication. [ENC]

The following figures illustrate the values of MD5 check sums in connection with this forensic examination. The first shows the MD5 check sums directly after the completion of the initial copy from the target system. Figures 2 and 3 show the MD5 check sums as calculated and displayed in the Autopsy forensic browser. As the numbers are a little difficult to read in the screen shots of the forensic browser image, the file that contains the MD5 check sums, md5.txt was displayed in an additional screen shot (Figure 4). In Figure 5, the screen shot shows the MD5 check sums at the completion of the forensic examination. This serves to illustrate that the process of examining the file systems did not damage the images, and that the data integrity of the images was preserved throughout the process. The images were cropped using Adobe Photoshop in order to better fit them into the report. Only blank space was cropped from the images.

Figure 1: MD5 check sums following copy



```
root@LinuxForensics:/forensics/41994/41994/images
File Edit View Terminal Go Help
[root@LinuxForensics images]# md5 dd.img.p1 dd.img.p2
f1a0253e1c07063f08cf5c944c0b8c9a      dd.img.p1
9995b8ea777a35d91f48c790073c3b99      dd.img.p2
[root@LinuxForensics images]#
```

Figure 2: MD5 checksum of C:\ partition

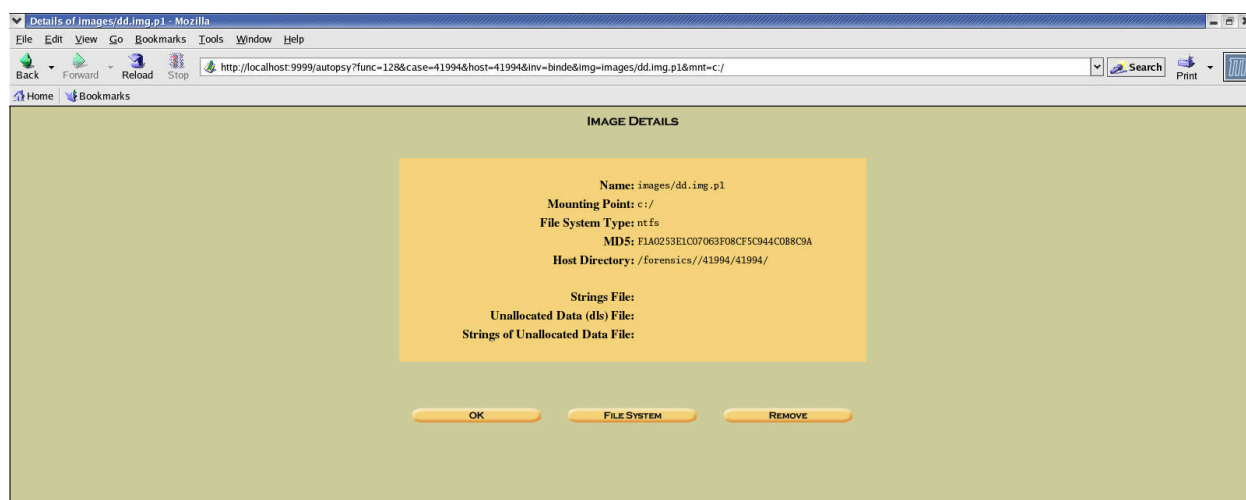


Figure 3: MD5 checksum of D:\ partition

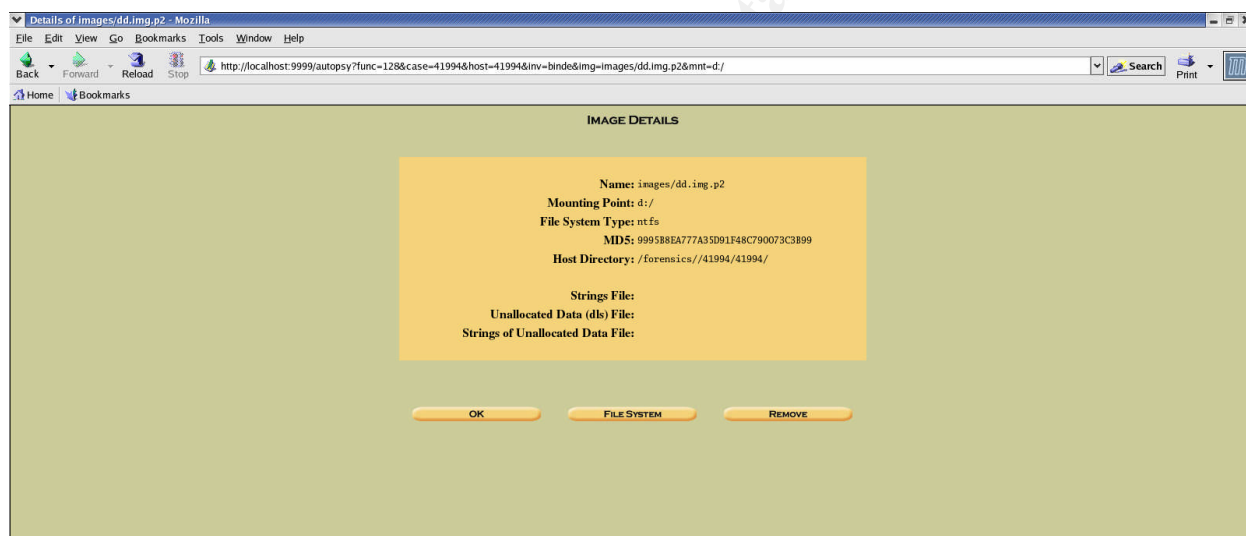


Figure 4: Display file with MD5 checksums

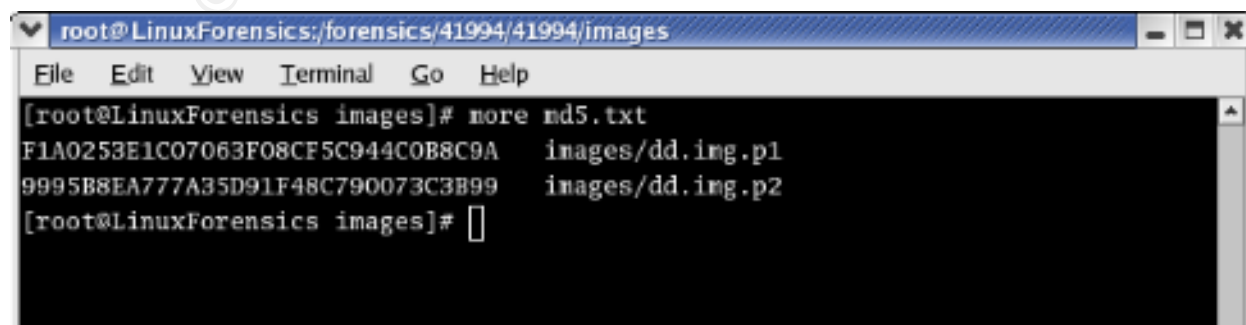
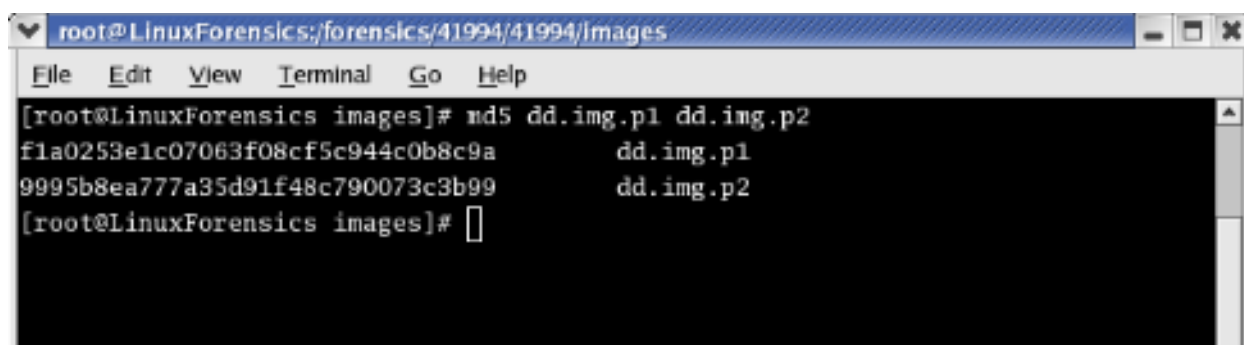


Figure 5: Final MD5 Checksum

A screenshot of a terminal window titled 'root@LinuxForensics:/forensics/41994/41994/images'. The terminal shows the command 'md5 dd.img.p1 dd.img.p2' being executed. The output displays two lines of MD5 hashes: 'f1a0253e1c07063f08cf5c944c0b8c9a dd.img.p1' and '9995b8ea777a35d91f48c790073c3b99 dd.img.p2'. The prompt '[root@LinuxForensics images]# ' is visible at the bottom.

```
root@LinuxForensics:/forensics/41994/41994/images
File Edit View Terminal Go Help
[root@LinuxForensics images]# md5 dd.img.p1 dd.img.p2
f1a0253e1c07063f08cf5c944c0b8c9a dd.img.p1
9995b8ea777a35d91f48c790073c3b99 dd.img.p2
[root@LinuxForensics images]#
```

Media and Timeline Analysis

Although these two areas are listed separately in the assignment, it was decided to combine them for the sake of a clearer presentation. Due to the circumstances surrounding this incident and the concern about whether data was accessed, the timeline was created first. The study of the time line led the investigation through areas that are more typically part of the media analysis. However, it was decided that a chronological presentation would provide a better description and results that could be more easily reproduced by another investigator.

A new case was opened in Autopsy, the forensic browser. This was the software that was used in the SANS GCFA course. In addition to being familiar, the software is also free. The other major reason for using Autopsy was that no commercial tools were available in house.

The two images were loaded into the forensic workstation from the USB disk to which they had been copied from the target machine. It took about 9 hours for the copy to complete. The USB disk was labeled and preserved as a backup of the images. The tower was stored in a locked office with limited access.

After opening a new case in Autopsy and load the dd image file, the next step was to use the various tools under the Create File Activity Time Lines tab. The initial step is to use the Create Data File tab to collect data about the image file. The default file name for the output file is, appropriately enough, named body. The format of the body file is filename, file permissions, file size and date information, with the | (pipe) character as a field separator. This file provides a basis for extracting timeline information used for further analysis. It comprises 171,738 lines.

Using the File Activity Time Lines feature of Autopsy, a timeline of the entire disk image was created. The data is extracted from the body file. This analysis shows the time

stamps for creation, modification and access of files across the file system in a more easily human readable format than that of the body file. The entire timeline takes up more than 410,155 lines, so no attempt has been made to include the entire file. Portions of interest have been selected for comment display and comment. Further, there is the very real problem of making sure that any identifying information has been redacted, which is harder to do safely in the presentation of larger files. The timeline begins in 1985, which predates the sale of the hardware. Given that the oldest file carries a name that indicates it was created locally, the system clock may have been set incorrectly on one or more occasions during which files were created. Another possibility is that the files were restored from backup to an even older machine but the same files are still in use.

Another timeline file was created starting on April 23, 2004, the date on which the system was hacked, and continuing until the time that it was taken down and turned over for examination. This file contains over 58,000 lines. In order to narrow the field of search somewhat, the department was asked to further describe the files of interest within the directory that had previously been identified. These files had a "txt" extension and were located in the %systemdrive%\inetpub\Ftproot directory. This is supposed to be the ftp target directory under IIS [MIC5]. A list of more than 800 was forwarded for examination by the department. There is a batch job that runs at 3 AM and accesses or updates a large number of files. However, a significant number of files were accessed at 2:25 AM and shortly thereafter. There was no corresponding scheduled task that were explain why the files may have been accessed. The department noted that the update of files at 2:25 AM was outside of normal activity and drew attention to that set of files.

The following snip from the timeline file shows a few of these files, with the filenames redacted. The file names are eminently descriptive, which is an excellent practice, but which also prevents the public disclosure of the filenames. All of the files were access on Saturday, April 24 at about 2:25 AM. Note that the file protections would evidently allow anyone to read, write/modify, or execute the files (if they were executables instead of text files). The Autopsy forensic browser uses the file protection syntax characteristic of UNIX, in which the file protection is often portrayed with a string of 10 characters. The first character is set if the file has some special characteristic, the most common value being blank, for a plain file, and the second most common being the letter d, to indicate a directory. The next 9 characters are to be read in groups of three each – the first three refer to the rights of the owner, the second three refer to the rights of a group of users who hold the same rights in common, and the third group of letters refers to the rights that any user on the system would have with regard to the file. If there is a hyphen instead of a letter value, the particular right is turned off. The letter r refers to read rights, the letter w refers to write (create and modify) rights, and the letter x to the right to execute, or run, the program. In the examples below, the owner of the file has read, write and execute privileges over the files, since the first three letters, rwx, are all shown. The same applies to the group of users to which this may apply, and finally, all users on the system appear to have the same rights over the file. While this may be an artifact of the UNIX inclinations within Autopsy, which runs under Linux, there are a few

files that have more restrictive permissions displayed. Therefore, the recommendation will be made to the department to audit the file protections that are set to make sure that they are appropriate for the system.

The initial number is a file size and the final number refers to the cluster on the disk where the file is located. The letter a indicates that this is an access time, as opposed to a create time (shown by the letter c) or modify time (shown by the letter m).

```
Sat Apr 24 2004 02:25:56
      576 .a. -/rwxrwxrwx 0 0 30886-128-1
d:/inetpub/ftpboot/redacted1.txt
     80892 .a. -/rwxrwxrwx 0 0 56210-128-3
d:/inetpub/ftpboot/redacted2.txt
     20993 .a. -/rwxrwxrwx 0 0 56208-128-3
d:/inetpub/ftpboot/redacted3.txt
```

Using the sorter facility took approximately 72 hours to complete. The program categorized 41,704 files on C: Of those, 26,770 are allocated and 14,934 are unallocated (deleted). On D: 130,032 files were found. Of those, 124,142 were allocated, and 5,890 unallocated. The NIST National Software Reference Library contains hashes of known operating system and application software. This permits investigators to focus on unknown binaries and files, and will hopefully make the search space smaller. The file sizes are quite large, and Autopsy seems to permit only one NIST NSRL database to be loaded for comparison at a time. The operating system CD was chosen as the best choice, the others being application software, images and graphics, and non-English software. [NIST] The NIST hash database recognized 7,416 files as part of the Windows 2000 operating system on C: and 3,209 on D: Those on D: appear to be associated with the web browser and IIS. However, that analysis still left thousands of files to examine.

The search began for data and system events corresponding to the times that were identified to be of interest. The three event logs were located in the C:\WINNT\system32\config directory. Using the extract facility in autopsy, the files were then copied to a ZIP disk. The ZIP disk was then mounted on a Windows 2000 desktop system. The Event Viewer was started and each Event file on the ZIP disk was opened for examination. The Application Event Log, Security Event Log, and System Event Log were examined in turn.

At around the critical time period, the Application Event Log showed a service called ci starting up. A search of the Microsoft web site enabled this to be identified as the Content Indexing Service, also known as the Indexing Service. [MIC4]. This is a facility provided by Windows to keep track of data for quick(er) retrieval. It uses the file system changes recorded by the Change Journal and will run after "enough" changes to

the file system have taken place. The settings in the registry keys MaxPendingDocuments, MaxFreshCount, and MaxFreshDeletes govern how soon the service will start up. It also runs after the system is booted. The number of minutes is shown in the registry key StartupDelay. The value for the system was left at the default value of 8 minutes. The indexing (event 4137 in the Application Events Log) took place at different times throughout the day, but most often at around 7 or 8 PM. It ran once a day, but oddly enough not on Sundays.

The StompLastAccessDelay registry entry is set for 7 days in the registry, so the last access time will get updated if the file was accessed within the last 7 days. The thinking behind this has to do with incremental backups. The idea is that if a file is not accessed within the past 7 days, it should not be copied in a backup operation – unless, of course, a full backup of all files is performed. It should be further noted that access time is not updated more than once every hour on a Windows system for performance reasons.

The log showed that the system rebooted unexpectedly at 2:18 AM. Since the Indexing Service runs 8 minutes after startup time, this explains why a large number of files were updated at about 2:25 AM.

In order to determine the value of the registry entries, the file C:\WINNT\system32\config\system was located and the extract facility of Autopsy used to copy the file to a ZIP disk, which was duly transferred to another Windows 2000 desktop system. On that system, a demo version of the Access Data Registry Viewer was installed to load the registry and examine the entries. This program is part of the AccessData Forensic Toolkit, often shortened to FTK. [ACC]

Netflow logs were also made available for examination from the 24 hour period beginning just after midnight on April 23, and extending until the next midnight. The Cisco netflow logs show accesses into and out of the network at the hand off to the outside internet. They do not show flows between devices within the network. The logs show an FTP access from a site in Germany at 7:53 PM and an exchange of about 8KB of data.

The source and destination IP addresses have been redacted through replacing them with the letter X for each quad in the IP address. Next follows either the port, or the range of ports. This varies as to whether the connection was incoming or outgoing. The first line and last line are incoming connections, and the two middle lines are outgoing connections. (This would be more obvious had it not been necessary to redact the IP addresses). FTP uses port 20 for control and port 21 for the actual transfer. The next field of interest show s the number of bytes transferred. This is the 7th field in the display. The last two fields refer to the number of seconds since midnight, December 31, 1969, Universal Time, and the present time. This is an artifact of the UNIX operating system. When the strings are translated with the help of a tool such as Decode Date (written by Craig Wilson and freely available from the <http://digital-detective.co.uk> web site), the timestamps translate to 7:53 PM on Friday, April 23,

2004. This was identified as a critical time since it corresponds to the creation time of the tools that were left behind by the hacker.

```
X.X.X.X|X.X.X.X|20|1K_9K_Port|TCP-FTP|652|836332|17|1082764377|1082764396
X.X.X.X|X.X.X.X|1K_9K_Port|20|TCP-FTP|496|20116|18|1082764377|1082764393
X.X.X.X|X.X.X.X|1K_9K_Port|21|TCP-FTP|86|4429|1|1082764376|1082764397
X.X.X.X|X.X.X.X|21|1K_9K_Port|TCP-FTP|49|3851|1|1082764381|1082764397
```

The remainder of the netflow logs showed accesses predominantly to ports 25 and 113 and 443. This would correspond with the SMTP activity found in the application event log. There is no ready explanation for the access to port 113 (the identd port). The activity on port 443 appears to line up with accesses to the web pages. The security bulletin MS04-011 indicates that the TCP ports 135, 139, 445 and 593 are vulnerable and should be blocked at the firewall. [MIC7] but the netflow logs for April 23 did not show activity on these ports. The web server logs from April 20 onwards were examined, especially for the evening of April 23. Nothing out of the ordinary was noted. There were no flows that would incontrovertibly indicate a successful login from a local outside of the network. Nor were there any flows that were indisputably the work of hackers, as opposed to the normal business of the host. This seems odd, as the intruder would most likely have wished to follow up after loading the root kit. It is possible that the activity was not logged in netflow. Another possibility is that the intruder accessed the system from within the University network, from another host that had already been compromised. This would not show up in the netflow logs. A third possibility is that the remote administration activity cannot reliably be identified from the netflow logs. Since the system unexpectedly rebooted at 2:18 AM on Saturday, April 24, it would appear that the intruder activity continued after the installation of the tools via ftp.

The FTP logs were located in C:\WINNT\system32\LogFiles\MSFTPSVC1. From the time stamps, it appears as if the files are rotated at 8 PM daily and reflect activity from the day before. The log files from April 20 onwards were examined. The file transfers appear routine for the most part. The userids were identifiable as legitimate and came from appropriate corresponding network locations. Again, the exact information has been redacted in this instance.

Peering more closely at the time stamps for the files from the time during which the intrusion toolkit was transferred, some distinctly odd characteristics are noted. The file for April 24 was created, access and modified on April 23. None of the other ftp log files were created in the future. There may or may not have been some intentional modification of the log files in order to muddy the (investigative) waters. The fragments from the listing of the ftp logs appear next.

Fri Apr 23 2004 20:00:00

m.c c:/WINNT/system32/LogFiles/MSFTPSVC1/ex040423.log

m.c c:/WINNT/system32/LogFiles/W3SVC1/ex040423.log

Fri Apr 23 2004 20:01:59

m.c d/drwxrwxrwx c:/WINNT/system32/LogFiles/MSFTPSVC1

m.c c:/WINNT/system32/LogFiles/MSFTPSVC1/ex040424.log

Fri Apr 23 2004 21:18:31

.a. c:/WINNT/system32/LogFiles/MSFTPSVC1/ex040424.log

.a. c:/WINNT/system32/LogFiles/MSFTPSVC1/ex040423.log

There was some excitement in the early morning hours of April 20, when there was a concerted but rather clumsy effort to guess a password for ftp access. The majority came from the same IP address, represented as X. X. X. X and also in Germany, but different than the one that successfully deposited a root kit on Friday, April 23. The details appear below, with the actual IP addresses redacted. In order to present the complete log fragment, it has been included without being framed by a table in order to allow it to flow to the next page.

```
02:28:16 X.X.X.X [1478]USER ftp 331
02:28:16 X.X.X.X [1478]PASS - 530
02:28:16 X.X.X.X [1479]USER ftp 331
02:28:17 X.X.X.X [1479]PASS ftp 530
02:28:17 X.X.X.X [1480]USER ftp 331
02:28:17 X.X.X.X [1480]PASS ftp12 530
02:28:18 X.X.X.X [1481]USER ftp 331
02:28:18 X.X.X.X [1481]PASS ftp123 530
02:28:18 X.X.X.X [1482]USER ftp 331
02:28:18 X.X.X.X [1482]PASS 123 530
02:28:19 X.X.X.X [1483]USER ftp 331
02:28:19 X.X.X.X [1483]PASS 12345 530
02:28:23 X.X.X.X [1484]USER ftp 331
02:28:23 X.X.X.X [1484]PASS 1 530
02:28:23 X.X.X.X [1485]USER ftp 331
02:28:27 X.X.X.X [1485]PASS 1234 530
02:28:27 X.X.X.X [1486]USER ftp 331
02:28:27 X.X.X.X [1486]PASS 123456 530
02:28:28 X.X.X.X [1487]USER ftp 331
02:28:28 X.X.X.X [1487]PASS 1234567 530
02:28:28 X.X.X.X [1488]USER ftp 331
02:28:29 X.X.X.X [1488]PASS 111 530
02:28:29 X.X.X.X [1489]USER ftp 331
02:28:29 X.X.X.X [1489]PASS admin 530
```

```

02:28:31 X.X.X.X [1490]USER ftp 331
02:28:31 X.X.X.X [1490]PASS passwd 530
02:28:31 X.X.X.X [1491]USER ftp 331
02:28:32 X.X.X.X [1491]PASS password 530
02:28:32 X.X.X.X [1492]USER ftp 331
02:28:32 X.X.X.X [1492]PASS pass 530
02:28:33 X.X.X.X [1493]USER ftp 331
02:28:33 X.X.X.X [1493]PASS super 530
02:43:53 X.X.X.X [1477]closed - 421
10:40:20 Y.Y.Y.Y[1494]USER anonymous@ftp.microsoft.com 331
10:40:20 Y.Y.Y.Y [1494]PASS - 530

```

The last two lines come from a different IP address and so are represented at Y.Y.Y.Y. The actual IP address has been redacted, but it most assuredly was not Microsoft.com nor connected with that company in any (obvious) way.

The next unusual event takes place on about 2 minutes after the regular file rotation on April 24. The entries after 2:25 AM are corrupted and appear to have been replaced by a binary file. This is about the same time as the unscheduled system reboot took place, and of course it could be that when the system rebooted unexpectedly, corruption was introduced. It may also have been an effort to cover up log entries that may have been helpful to tracking back to the person(s) responsible for the intrusion.

Here the timeline that closely relates to the April 23 intrusion has been formatted to fit on the page gracefully. Extraneous lines have been deleted.

The first event is the creation of the classes subdirectory at 7:52 PM.

```

Fri Apr 23 2004 19:52:37
56 m.c d/drwxrwxrwx 53-144-6 c:/WINNT/java/classes
48 m.c d/drwxrwxrwx 11812-144 c:/WINNT/java/classes/jar/scripts

```

Next, an ftp application is accessed. The log file for FTP does not show a data transfer at this time, although it is discernible in the netflow logs.

```

Fri Apr 23 2004 19:52:55

39696 .a. -/-rwxrwxrwx 570-128-3 c:/WINNT/system32/FTP.EXE

```

The reason why this transfer did not get logged became evident later. It was presumed at first that the executable file listed above was part of the Microsoft operating system distribution. However, it is not recognized in the NSRL database. Rather C:\WINNT\system32\irftp.exe was identified. See the NIST file identification information below. [NIST]

```
c:/WINNT/system32/irftp.exe
Image: /forensics//41994/41994/images/dd.img.pl Inode: 1004-128-4
MS-DOS executable (EXE), OS/2 or MS Windows
MD5: 56e4e9069611a1cd9d4b25f6a7ddc4ad
NSRL Database
```

```
c:/WINNT/system32/FTP.EXE
MS-DOS executable (EXE), OS/2 or MS Windows
Image: /forensics//41994/41994/images/dd.img.pl Inode: 570-128-3
MD5: edc5a5f84071bf78e6f1d325df7192ad
```

The mystery deepened further, as the file was listed as being modified on June 19, 2003 and changed on April 27, 2004 at 10:23 AM, after the machine had been disconnected from the network permanently. An examination of the current binary did not reveal the origin. The registry does not list the executable as one that gets initialized at startup time. There is not enough information left to draw a firm conclusion about this executable.

In the next few seconds, the creation/modification of the files that seem to be part of an intrusion tool set are shown. Recall that the file names were changed by the system administrator on Monday morning with the intent of preventing the further execution of the files. These events have been combined to one chart for the sake of grouping similar lines together.

```
Fri Apr 23 2004 19:52:57
```

```
183 m.. -/-rwxrwxrwx      13141-128-1 c:/WINNT/java/classes/jar/pnw32dll.bak
374 m.. -/-rwxrwxrwx      13235-128-1 c:/WINNT/java/classes/jar/ole32dll.bak
```

```
Fri Apr 23 2004 19:52:59
```

```
90112 m.. -/-rwxrwxrwx      13247-128-3 c:/WINNT/java/classes/jar/AdmDll.dll.bak
758 m.. -/-rwxrwxrwx      3268-128-3 c:/WINNT/java/classes/jar/secbat.bak
253 m.. -/-rwxrwxrwx      13274-128-1 c:/WINNT/java/classes/jar/endbat.bak
```

```
Fri Apr 23 2004 19:53:02
```

```
65536 m.. -/-rwxrwxrwx      13287-128-3 c:/WINNT/java/classes/jar/psinfoexe.bak
```

```
Fri Apr 23 2004 19:53:07
```

```
769536 m.. -/-rwxrwxrwx 13300-128-3 c:/WINNT/java/classes/jar/svchostexe.bak
```

```
Fri Apr 23 2004 19:53:09
```

```
66048 m.. -/-rwxrwxrwx      13309-128-3 c:/WINNT/java/classes/jar/infoexe.bak
361214 m.. -/-rwxrwxrwx 13328-128-3 c:/WINNT/java/classes/jar/nvsvcexe.bak
```

Several of these executables were harvested using the extract facilities within Autopsy, and transferred to a test Windows 2000 laptop for analysis. The executables were loaded into the Ollydbg program. Unfortunately, this examination did not yield additional helpful information about the function of the programs. [YUS]

Next, we see the use of the ipconfig.exe program, normally used for viewing or setting network characteristics. This takes place a full minute after the creation of the C:\WINNT\java\classes directory. This leads one to believe that this command was typed by hand rather than completely run from a script.

```
Fri Apr 23 2004 19:53:39
```

```
35600 .a. -/-rwxrwxrwx      944-128-4 c:/WINNT/system32/ipconfig.exe
```

Finally, the creation of a new directory, ug31 is noted. This is odd in that the files have nothing in them, and there is no indication that they were later modified.

```
Fri Apr 23 2004 20:05:01
```

```
0 .ac -/-rwxrwxrwx 15654-128-666 d:/ug31/autoexec.bat
6 ..c d/drwxrwxrwx 15650-144-9 d:/ug31
0 .ac -/-rwxrwxrwx 15653-128-667 d:/ug31/autoedxec.bat
```

Going deeper, here are the contents of the secbat.bak file. The original name of the file was sec.bat, but it was renamed by the system administrator. It appears to be a script that shows some of the actions that were taken by the hackers, and the capabilities of the root kit. If the comments are to be believed, it appears as if an RPC buffer overflow was part of the exploit. A diligent search was made to find the dcompatch.exe file but it could not be identified. It looked as if it might be a file of great interest in terms of understanding the actual intrusion. The file was sought by using grep to search the body file created by the Autopsy Forensic Browser.

```
echo : TheCabal has you>install.log
echo :>>install.log
echo : Please stand by while the system is being prepared and r00ted...>>install.log
echo :>>install.log
echo :>>install.log

echo : Securing service files...>>install.log
attrib +s %windir%\java\classes\jar\
attrib +s %windir%\java\classes\jar\javadoc
attrib +s %windir%\java\classes\jar\scripts
echo : Service files secured...>>install.log
echo :>>install.log

echo : Closing additional exploits...>>install.log
dcompatch.exe
echo : RPC Buffer Overflow Exploit Closed>>install.log
echo : Additional exploits closed!>>install.log

echo :>>install.log
echo : The system has been secured and r00ted by TheCabal>>install.log
echo :>>install.log
echo : Have a nice day!>>install.log
```

Also left for the pleasure (and frustration) of the forensic examiner was the following script, endbat.bak (originally end.bat but renamed by the system administrator). This script lists a number of files that are deleted upon cleanup. A search was made for these files, but they were not located in the forensic image.

© SANS Institute 2004, Author retains full rights.


```
@echo off
del b.bat
del ms.bat
del uptime.exe
del fport.exe
del ports.exe
del CommonDlg32.dll
del nvsvc32.dll
del raddrv.dll
del radmin.reg
del quark.exe
del dcompatch.exe
del temp.txt
del ports.txt
del kill.exe
del DryGz.txt
@echo off
```

Among the key features of the Famatech Remote Administration program are the following:

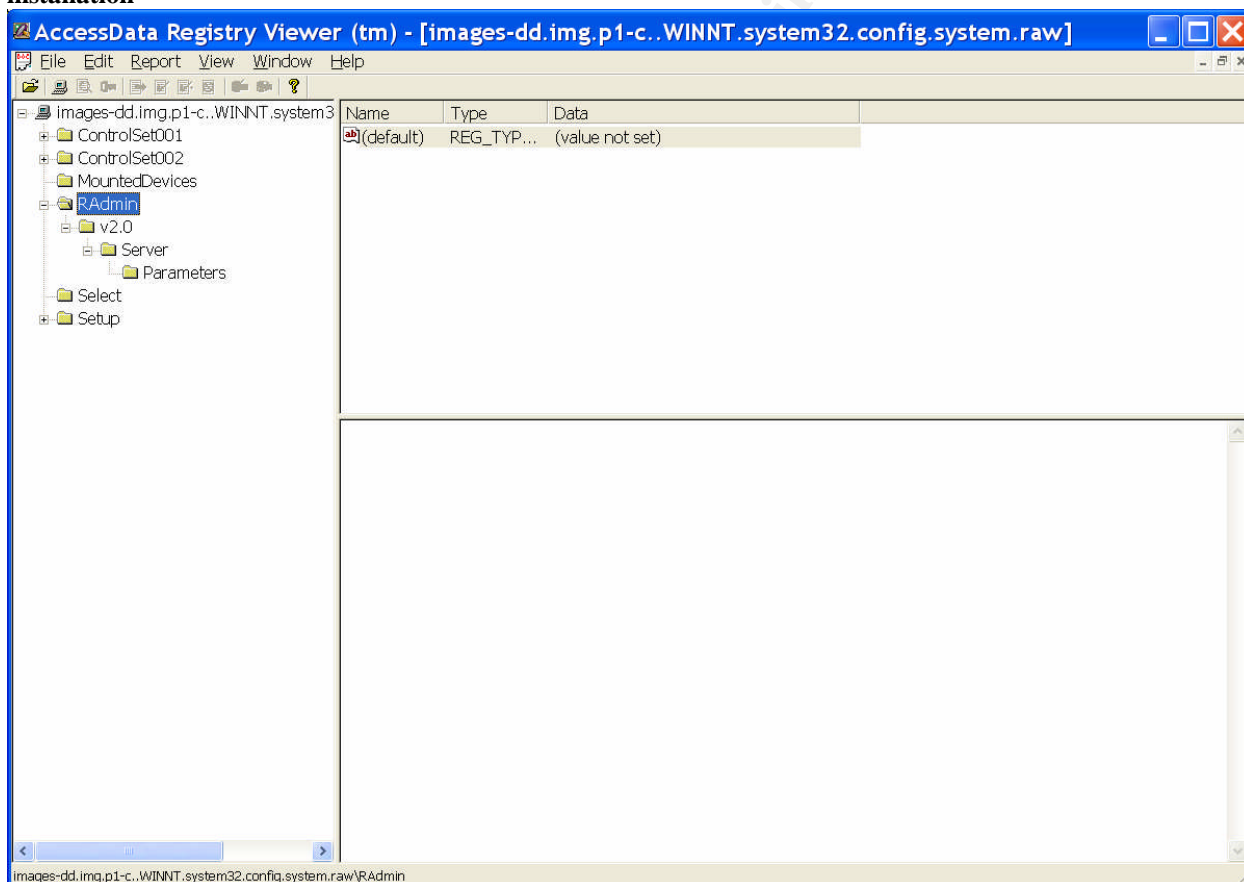
- supports multiple connections, on both client and server sides
- controls more than one remote computer
- transfers files to and from the remote computer
- permits remote shutdown
- data streams are encrypted

This program clearly has many advantages for remote administration and control, whether by the legitimate system administrator or a hacker. The help file that comes bundled with the program specifies that port 4899 is the default port. This could, perhaps, be changed to another port number. The netflow logs did not show activity that could be pinpointed as originating on port 4899. [FAM]

The Famatech Remote Administration program was downloaded from the web site for testing. Version 2.1 is the current version, and version 2.0 is no longer available. A telephone call to the company to request the MD5 check sum is not expected to be successful in obtaining the information. The application was downloaded and then transferred to a test Windows system. A comparison was made between the files found in C:\WINNT\java\classes\jar. The MD5 checksums are reproduced below for the entire distribution of Remote Administration version 2.1. The checksum for the first file, AdmDll.dll, is the only one that matches up.

c915181e93fe3d4c41b1963180d3c535	AdmDll.dll
800c6ee5dec7a0e71a6640a08d452091	help.cnt
6ba2449b06128cf16fcf29d5a90d9f3e	help.hlp
b04f8b657227128a86744d1898491a74	license.txt
d212242420f53333c2d764acffd1750e	raddrv.dll
c9d106166897d65573b9b326b1744cd7	radmin.exe
29a54631ea18c47269199cd18eeea343	readme.txt
21ebf2467cf6270df7b0077fc2cbf243	r_server.exe
b83429c6f8335b63dd316bb83edaff23	uninstal.exe
579590edbd481e63042520ffd4ba0237	uninstal.ini
f6cf6abb534743a596093f170a8db96c	WhatsNew.txt

Figure 6: Registry keys showing Radmin installation



Some further trolling was done through the registry to find keys of interest to services and executables that are automatically started up. This is illustrated in Figure 7. For

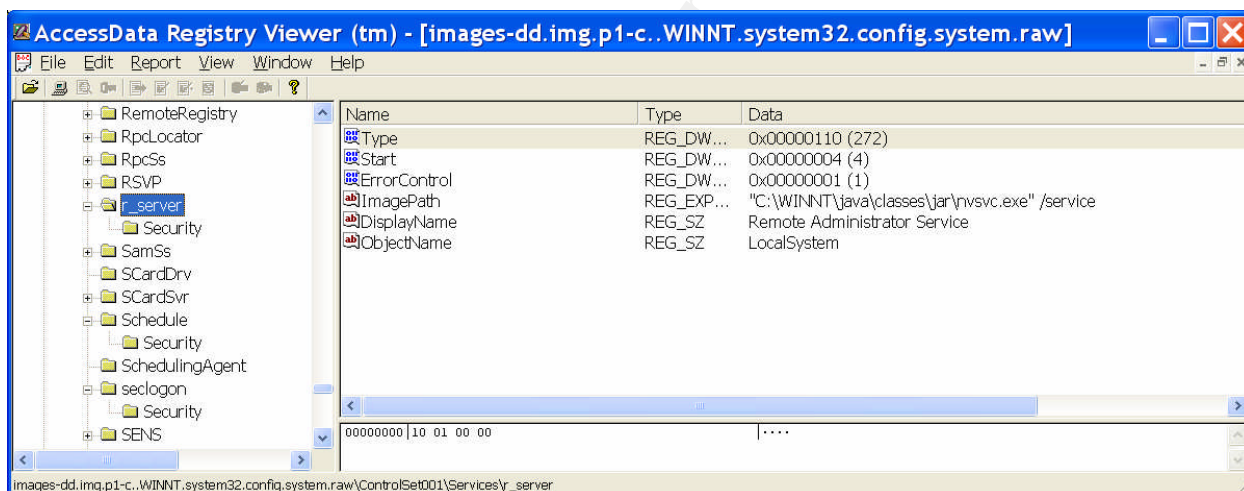
example, the remote server for Remote Admin starts up at boot time as shown below, with the command:

```
C:\WINNT\java\classes\jar\nvsvc.exe /service
```

This entry also tends to confirm the report of the system administrator with regard to the original file names. This file is called r_server.exe in the distribution that was downloaded from the Famatech website, but was called nvsvc.exe in the copy that was installed by the intruder, and was renamed by the system administrator to nvsvcexe.bak on Monday morning, April 26, in an attempt to secure the machine against further damage.

Given that the program was found running on Monday morning with the icon in the system tray, and the supporting information gleaned from registry entries and one MD5 checksum, it seems reasonable to conclude that the tool kit did in fact include the Famatech Remote Administration program.

Figure 7: Remote Service startup entry in registry



The registry entries in autorun.exe appear to refer to standard application startup programs;

- PagisPro 2.0
- OfficeAccounting1.0
- CDKakumeiVirtual
- EasyCDCreator 3.x
- PagisPro 3.0
- ZipMagic98
- WindowsNT4.0

- SNAServer
- VirusBuster9xforNT
- SMSServer
- CalendarCreator5

However, it is not clear why all of these programs would need to be installed on a server. For example, the VirusBuster program should have been superseded by the site licensed anti-virus software product. The role of an Office Accounting program on a dedicated server is unclear as well. It would seem as if some rather old programs that ran under Windows 9x versions are still in place. That would be expected in a system that had been installed 4 years previously.

Other locations were checked for additional startup files. These included C:\WINNT\win.ini and C:\WINNT\system.ini. Neither had anything of interest. Another file, C:\WINNT\wininit.ini was mentioned as another location for startup files, but it did not exist on this system. [SKO]

A lead was followed up on a string search that is described in greater detail in the section of this paper where the string searches are described and the logic that brought attention to the C:\WINNT\system32\wbem\mof\USR32\WEB\BACKUP\ directory. The means by which this directory came to the investigative fore are almost of a serendipitous nature. The full details have been preserved for the section on string searches.

Opening the directory in the FILE ANALYSIS screen of the Autopsy Forensic browser, the tree was traversed to see what might be there besides the se.txt file. The se.txt file itself was not of much interest. It appeared to be a template for displaying system and or program uptime. However, the directory contained touch.exe, which from the comments, is a program to update the access time on a file. It appears to be a port of the well known UNIX utility for updating access time on a file. A display of the strings from the svchost.exe file shows that it seems to be an FTP server called ServU, which seems to be a popular choice in the hacker community. These files were viewed in the FILE ANALYSIS screen of Autopsy, which permits files to be displayed, exported, and also can apply the Linux strings command, which is helpful for teasing out information about executable files.

The chilling factor about these particular files is that the creation dates on the files are from April 22, the day previous to the supposed hack. The file below would seem to indicate that three FTP servers were all listening, and those on non-standard ports. (The X.X.X.X refers to the IP address of the host that is the subject of this analysis, which has been redacted for the protection of the institution.) This directory would not appear to be a standard location for FTP log files, and the presence of other files clearly associated with the intrusion further arouses suspicion. Note the incorrect spellings for Microsoft as Micosoft. These directories and files were all created at 1:06 PM on Thursday, April 22.

```
Thu 22Apr04 13:06:06 - IP / TCP Services v5.0 - (5.0.0.4) - Copyright (c)
1995-2004 Micosoft, All Rights Reserved - by Bob Myerson
Thu 22Apr04 13:06:06 - Micosoft is an affiliate of Microsoft Corp, Inc.
Thu 22Apr04 13:06:06 - Using WinSock 2.0 - max. 32767 sockets
Thu 22Apr04 13:06:07 - PROBLEM: Unable to load the SSL/TLS libraries
(SSLEAY32.DLL and LIBEAY32.DLL) - No SSL support
Thu 22Apr04 13:06:07 - FTP Server listening on port number 206, IP X.X.X.X,
127.0.0.1
Thu 22Apr04 13:06:07 - FTP Server listening on port number 1926, IP X.X.X.X,
127.0.0.1
Thu 22Apr04 13:06:07 - FTP Server listening on port number 54445, IP
127.0.0.1
Thu 22Apr04 13:06:07 - Valid registration key found
```

Thus, a simple string search from which little of interest was expected, led back to the discovery of a hack previous to the one originally noted by the system administrator.

Three hack attempts have been discovered, two of them apparently successful. The Event Logs were scrutinized again with the hope of uncovered some corroborating information, focusing on the three critical time periods that have been identified so far:

1. April 20. 02:28 AM and 10:40 AM – brute force password attacks
2. April 22 13:06 – startup of three rogue FTP servers
3. April 23 7:50 AM – transfer of Remote Administration toolkit via FTP

The Application Event Log was examined again around these critical times, but there appeared to be no events of interest, and in fact, few events at all in those time periods.

Next the Security Event Log was examined around these critical times. Logins and logouts around the time period of event #1 were all system related, such as scheduled tasks starting up. Similarly for event #2. The third event window was in the evening, the backup user showed two logins. The system administrator said that was normal for the system. The scheduleme task also ran normally.

The Microsoft web site [MIC6] provides the following explanation of login types:

- 2 Interactive
- 3 Network
- 4 Batch
- 5 Service
- 6 Proxy
- 7 Unlock Workstation

The same article on “Auditing User Authentication” provides a complete listing of login event numbers and a description of each, too lengthy to reproduce here, but essential to examining the Security Events Log.

There were network logins by privileged users during this time period. They do not show up in the netflow logs for those time periods, therefore it is presumed that the logins were performed from another site within the Rutgers network, or perhaps by “hopping” through an inside host. The sysadmin administrator routinely checks on system status during evening hours. As a precaution, all passwords were changed on the replacement system as a precaution. Again, there was nothing that could definitively be marked as suspicious. There appeared to be valid explanations for the system activity.

Finally, the System Event Log was examined. Upon discovering the intrusion, the system administrator changed passwords and installed hotfixes that had previously been set aside. All the other hosts had already been updated. This server, the most critical, was delayed until it could be determined that the hotfixes were stable enough for installation.

On Monday morning, April 26, the following hotfixes were installed:

- KB835732: Microsoft Security Bulletin MS04-011 [MIC7]
- KB828741: Microsoft Security Bulletin MS04-012: Cumulative Update for Microsoft RPC/DCOM [MIC8]
- KB837001: Microsoft Security Bulletin MS04-014: Vulnerability in the Microsoft Jet Database Engine [MIC9]
- Q828026: Critical Update for Windows Media Player [MIC10]

The first bulletin especially covered a number of important vulnerabilities. However, both MS04-011 and MS04-012 warned that remote code execution was a possibility in connection with the vulnerabilities. The liberty will be taken to focus attention on the LSASS vulnerability (CAN2003-0533). A buffer overrun vulnerability in LSASS could allow remote execution of arbitrary code on a system. The bulletin goes on to explain that an attacker could take complete control. A reference web page goes on to give a definition of buffer overrun that is worthwhile to quote:

Buffer Overrun:

An attack in which a malicious user exploits an unchecked buffer in a program and overwrites the program code with their own data. If the program code is overwritten with new executable code, the effect is to change the program's operation as dictated by the attacker. If overwritten with other data, the likely effect is to cause the program to crash. [MIC11]

The bulletin goes on to note that Windows 2000 can be remotely attacked by an anonymous user. [MIC7].

Fixing the information about the newly announced set of vulnerabilities firmly in mind, return again to the System Event Log. On Friday, April 23 at 3:59:51 PM, the security package NTLM generated an exception, and the package was disabled. The source of the event was LsaSrv and the Event ID was 5000. The event is described as an access violation, and the recommended resolution is to restart the computer. The Local Security Authentication Subsystem (LSASS) process is also tied to the LsaSrv service. [MIC12] At 8:27:31 PM, LsaSrv logged an additional error, this when an exception was generated in the Unified Security Protocol Provider security package. The third error was generated at 12:49:22 PM, another exception generated by NTLM. The Security Event Log lists many Anonymous logins throughout the life of the log. If the exercise of the exploit is logged in this fashion, the exploit seems to have been freely taken advantage of. Given the vivid description in the MS04-011 security bulletin, it would seem that an exploit of the LSASS vulnerability was a major factor in the April 23 intrusion. However, the methodology of the April 22 intrusion remains unguessed.

To complete the timeline and media analysis, the dates of major system updates are included. The simple expedient of checking the timeline file for access date changes in the C:\WINNT\ServicePackFiles directory was used to identify the time periods. As well over 5,200 events were identified, the decision was made not to include the entire time line, but rather to choose a line containing the timestamp from each major update. In the interests of fitting the output on the page, the cluster number, file protection, file size and so forth have been edited out so that only the date and the first file name remain. The resulting output follows.

```
Fri Jul 07 2000 15:05:01 c:/WINNT/ServicePackFiles/i386/xmldso.cab
Mon Oct 16 2000 20:51:26 c:/WINNT/ServicePackFiles/i386/vbajet32.dll (deleted-realloc)
Tue Mar 06 2001 08:28:01 c:/WINNT/ServicePackFiles/i386/wms41.cab
Wed Jan 23 2002 15:08:01 c:/WINNT/ServicePackFiles/i386/wlbs.hlp
Wed Sep 25 2002 21:38:47 c:/WINNT/ServicePackFiles/i386/wuauhelp.chm
Wed Dec 11 2002 12:25:17 c:/WINNT/ServicePackFiles/i386/msxmlr.dll (deleted-realloc)
Thu Jan 08 2004 09:17:51 c:/WINNT/ServicePackFiles/i386/halborg.dll (deleted)
Sun Feb 22 2004 18:45:16 c:/WINNT/ServicePackFiles/i386/lsasrv.dll (deleted)
Mon Mar 08 2004 11:19:03 c:/WINNT/ServicePackFiles/i386/apcompat.inf
Mon Apr 05 2004 21:23:40 c:/WINNT/ServicePackFiles/i386/agpcpq.sys (deleted-realloc)
```

It would appear that updates were made along with or shortly after the initial installation of the system. The shipping date of the system was June 2, 2000, so the first set of updates on July 7, 2000, seem to be around the time that the system most likely would have been installed. The next set of updates occur in a timely fashion, only three months later in October, 2001. It is five months before the next set are installed in March, 2001, and a very lengthy 10 months before the January, 2002 install of the next set. The lagged schedule persists, waiting 8 months until September, 2002. Only four months elapsed before the December, 2002 updates, then over a year – a full thirteen

months, before another set. Since the opening of the year 2004, the patching and updating has improved to a monthly basis, as updates were installed in every single month of the year. This is a considerable overall improvement in the patch management of the system.

Recover Deleted Files

Within Autopsy, recovery of deleted files is easy. While in the File Browsing Menu, there is a button labeled "All Deleted Files". Clicking on the button reveals the names of all files on the file system that were flagged as deleted. Underlined filenames are recoverable if the files have not already been reallocated (those that were reallocated are so labeled). It is possible to display the file, run the strings program against it, or export it to a file outside of Autopsy for examination with other tools.

No deleted files of interest could be located to support this investigation, though not through the lack of effort to identify files that were expected. In order to complete the assignment requirements, a file was recovered in order to demonstrate the technique. A file without sensitive information needed to be located as well, which left out many more attractive possibilities. The file shown below is a fragment from C:\Program Files\Oracle\jre\1.3.1\lib\jvm.hprof.txt:

```
Copyright 1998 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto,
California, 94303, U.S.A. All Rights Reserved.
```

```
WARNING! This file format is under development, and is subject to
change without notice.
```

```
This file contains the following types of records:
```

```
THREAD START
```

```
THREAD END      mark the lifetime of Java threads
```

```
TRACE
```

```
represents a Java stack trace. Each trace consists
of a series of stack frames. Other records refer to
TRACES to identify (1) where object allocations have
taken place, (2) the frames in which GC roots were
found, and (3) frequently executed methods.
```

```
HEAP DUMP
```

```
is a complete snapshot of all live objects in the Java
heap. Following distinctions are made:
```

This file also serves to illustrate that the host has Oracle installed along with the many other applications.

String Search

String searches were not a significant part of this particular forensic examination, although they can often be a gold mine of information and yielded some interesting results, for all that. Searches were performed for the IP address belonging to the German site from which the intrusion toolkit was transferred via FTP, but there were no hits. An attempt was made to search for all IP addresses, using a regular expression supplied by the Autopsy forensic browser. After running for over 12 hours, this search hung the system and was terminated.

The name “Jean-Loup Gailly” was found in the strings of the nvsvc.exe.bak file (the Remote Admin server executable, renamed from r_server.exe), so a search was performed for that as well. The name appears in the radmin.exe downloaded from the Famatech web site as well. Since the current version of the Remote Admin program is 2.1, and the version installed as part of the root kit is 2.0, this name helps tie the software together. [FAM] Because the name was found within the executable, it was thought to be a good candidate for a further strings search. It was found in c:\Program Files\Common Files\Microsoft Shared\VGX\vgx.dll. A Google search was performed and Mr. Gailly’s personal home page was quickly located at <http://www.teaser.fr/~jlqailly/>. Mr. Gailly is the author of the essential gzip compression utility. It would appear that he has established himself as an international presence in the community, and is unlikely to be connected with hacking or writing hacking code.

Since the moniker “TheCabal” was left in one of the batch files found in c:\winnt\java\classes\jar, it also seemed like a good string for which to search. It was found again in the batch file, of course. In addition, it was found in c:\pagefile.sys, the file where the memory contents are written. This would indicate that the batch file had at one point been run on the system, perhaps to attack another host.

Another search located “TheCabal” in a file that was not allocated to any meta data structures, but it was still able to be displayed in the Autopsy forensic browser. There was a note that a “Filler Entry exists in fs_data_run” but a search on the Internet did not prove fruitful for finding information as to the meaning of this description. The files are, perhaps, outside of the “normal” file system structure. In any case, the file appeared to be a set of parameters for the running of the intrusion toolkit. Two short segments from the file follow:

```
[DOMAINS]
Domain1=0.0.0.0||7748|Admin|1|0|0
Domain2=0.0.0.0||1149|Leechers|2|0|0
Domain3=0.0.0.0||21|Cabal|3|0|0

User1=thea|1|0
```

```
User2=scan|1|0
DirChangeMesFile=C:\WINNT\java\classes\jar\pnw32.dll
SignOn=C:\WINNT\java\classes\jar\ole32.dll
DirChangeMesFile2=C:\WINNT\java\classes\jar\pnw32.dll
```

```
[USER=tools|3]
Password=fp14BEA1846064E9F63196414FE6B5957C
HomeDir=c:\system volume information\.log
RelPaths=1
TimeOut=600
Access1=c:\
```

It is interesting that this file ties back to the location where the Remote Administration kit was located, C:\WINNT\java\classes\jar. Further, the mention of “leechers” hardly seems to be characteristic of commercial software. Therefore, a further search was done on the string “leechers”. This uncovered another configuration file with similar password-like entries for “leechers” and so forth. However, the configuration file pointed to an entirely different directory for the DirChangeMesFile parament mentioned above. Instead of the directory C:\WINNT\java\classes\jar, the DirChangeMesFile is the somewhat lengthy C:\WINNT\system32\wbem\mof\USR32\WEB\BACKUP\se.txt. While it has the appearance of a system file of some sort, that was true of the other directory location as well. This seemed like a lead worth pursuing. The results of the further investigation have already been treated in the section that comes the Media and Timeline Analysis.

Conclusions

It appears that the Content Indexing Service was responsible for updating the access time on critical files (as well as non-critical files) throughout the system. It is possible that the intruder accessed the files but the later action of the Content Indexing Service wiped out evidence that the files had been accessed. This was a primary concern of the victim department, since the data was sensitive.

Three different intrusion attempts were identified, all taking place within the week. The first was a brute force password attack that did not appear to succeed. Secondly, a deposit of inappropriate files were located with a date of April 22. These files appeared to live outside of the normally accessible disk area. Finally, an intrusion toolkit was dropped on the system at about 7:50 PM or so on April 23, which included a remote administration program. The Remote Administration icon was showed in the system tray after the first reboot.

Given the April 13 announcement about the LSASS vulnerability and system errors congruent with a buffer overrun, it is surmised that the April 23 intrusion was an exploitation of that vulnerability. Files deposited on the previous day have the string

“leechers” in common with files deposited on April 23. The exploit used on the previous day appears to have been different and was not identifiable. Further, as no network activity was logged other than an FTP transfer of an intrusion toolkit, there is a possibility that the intruder continued accessing the victim host from another machine within the University network.

A number of recommendations will be made to the department to improve security and help prevent future intrusions. The most critical suggestions are:

- Segregate services such as email, ftp and web access from each other.
- Isolate service machines from multiuser hosts
- Turn on firewall rules to permit only the necessary services.
- Patch systems promptly.
- Update passwords on a regular basis.
- Verify that file ownership and file protections are set appropriately, especially for sensitive files in the ftp download area. .

Further experience and research are required in order to develop a set of processes and procedures for handling forensic investigations. The techniques used to copy disks must be sound, and the implications of choices (such as the use of reiserfs) must be clearly understood. In an ideal situation, a forensic image is collected under carefully controlled conditions aimed to prevent the unintentional alteration of evidence. It is important to recognize and identify possible evidence, document it, collect it, then package it and transport it safely and securely. A "jump kit" for evidence collection should always be at the ready and include the following components:

- documentation tools (felt tip markers, tags, paper pads, a digital camera),
- disassembly and removal tools (screw drivers, pliers, tweezers),
- packaging and transport tools (anti-static bags, bubble wrap, tape, sturdy boxes),
- data collection tools (cables, connectors, forensically wiped disk drives, floppy disks, forensic software)
- other useful items (gloves, a small flashlight, list of contact phone numbers).

Additionally, the perfect scenario includes a fast forensic workstation, multiple external disks, and a quiet lab in which to spend extended hours examining the image and following up on investigative possibilities. In this imaginary world, neither virus outbreaks, departmental reorganizations, nor other job responsibilities should be permitted to intrude upon the sanctuary work of the forensic examiner. While this may be possible, it is not probable. It is expected that many other things will happen sooner than this fond hope is realized.

Part 3 – Legal Issues of Incident Handling

This section of the paper presumes that Mr. John Price was distributing copyrighted material on publicly available systems and poses specific questions about applicable laws.

A. Without the specifics of the actions allegedly taken by Mr. Price, it is not possible to be certain as to which laws would apply. However, the Digital Millennium Copyright Act (DMCA) is a federal law in the United States that was enacted to govern online copyright infringement. It is the American implementation of an agreement reached in Geneva, Switzerland in October of 1971. It was enacted in 1998 and represents a major overhaul of copyright law in the United States. The act limits the liability of online service providers, a broad definition that would include large research universities. The requirement is to suspend the computer accounts of repeat offenders, and to designate a DMCA agent to keep records of violations. It may be possible to prosecute under state laws for damage to the computer equipment. Without more detailed information, it is impossible to tell whether or not that damage would reach the monetary threshold for federal statutes. However, the focus is on the issue of copyrighted material. New Jersey does not have separate statutes with regard to electronic copyright. [DMCA] Mr. Price may be individually responsible for his actions, although the institution may not be.

B. If the copyrighted material belongs to another institution, they would be obligated to contact the DMCA agent at our institution. If the copyrighted material is discovered internally there is no requirement to notify the DMCA, but it is often done anyway in practice. These are collectively known as the “take down” provisions of the act. The law has no requirement to contact the owner of the material. However, the material must be removed from public distribution should the copyright owner file a complaint. The law permits educational institutions to hold the material for the purpose of evaluating whether or not to purchase it. As it would be a prudent step, the material would be taken down from publicly available systems, whether there had been a complaint from the copyright holder or not. Educational institutions enjoy limitations on liability for possible DMCA violations, and exemption from fines and criminal penalties. While this protects the institution, it would not protect individuals within an institution who, for example, distribute copyrighted music in digital format. [DMCA]

C. If corporate counsel decides not to pursue charges against Mr. Price immediately, steps must be taken to preserve the evidence for a reasonable period of time, should the situation change. One means of determining a reasonable period of time would be to check on the statute of limitations. In New Jersey, that time period is 10 years. [NJ2] The DMCA specifies a 5 year statute of limitations [DMCA].

The material should be locked up in a secure facility with limited access by as few persons as possible. As far as is possible, any access to the facility should be logged.

A good effort is to put the materials in an envelope, seal it, and sign it across the flap. The chain of custody begins at the point when law enforcement steps in. However, good practices will aid the prosecution in making a good case.

D. If the investigation disclosed that Mr. Price was distributing child pornography, the following incident handling procedure is recommended:

- First, any further research by University technical staff would stop immediately.
- The equipment and data would be secured as quickly and expeditiously as possible to preserve it from unintended alteration. The area and equipment are now, for all practical purposes, part of a crime scene.
- Senior management (at the director level) would be advised of the situation. Another senior manager would be brought in, if necessary.
- University Counsel and the University Police would be contacted.
- The investigation, if any, would then belong to the University Police, who may chose to contact the local FBI office for assistance.
- Senior management would interface with University Public Relations for any publicity concerns as well as the appropriate office for disciplinary concerns. (This varies according to the status of the alleged perpetrator, whether a student, faculty member, staff member or graduate student).
- The technical staff would be reminded of the Non-Disclosure Agreement.

If possible, the image would remain on the monitor until a detective could arrive to make a first hand determination as to whether the image met the legal definition of child pornography. This would place the images in plain view of the officer; at least initially, a search warrant would not be necessary to begin an investigation.

The involvement of the technical staff in the investigation would be over as soon as criminal activity was detected. Law enforcement may have some technical questions pertaining to the actual forensic examination, but law enforcement officials would be in charge of any further investigation. Handling would vary so drastically because of the nature of laws against child pornography. Possession of a single image is an offense, not to mention multiple images or distribution. Hence the investigator needs to take extreme care to avoid any possibility of a false accusation of possession.

References

- [ACC] "Access Data Forensic Tool Kit"
http://www.accessdata.com/Product04_Overview.htm?ProductNum=04
(June 6, 2004).
- [BAC] Baca, Ernest "Knoppix Bootable CD Validation Study for Live Forensic Preview of Suspect's [sic] Computer" <http://www.linux-forensics.com/forensics/KNOPPIXValidation.pdf> (June 11, 2004).
- [CAR1] Carrier, Brian. "dd Acquisitions" *The Sleuthkit Informer* Issue #11, December 15, 2003. <http://www.sleuthkit.org/informer/sleuthkit-informer-11.html> (March 15, 2004)
- [CAR2] Carrier, Brian. Autopsy Forensic Browser.
<http://www.sleuthkit.org/autopsy/> (April 12, 2004).
- [CHU] Chuvakin, Anton. "Linux Data Hiding and Recovery".
LinuxSecurity.Com http://www.linuxsecurity.com/feature_stories/data-hiding-forensics.html (March 15, 2004).
- [COU] Courtney, Scott "An In-Depth Look at Reiserfs"
<http://www.linuxplanet.com/linuxplanet/tutorials/2926/4>. (June 10, 2004).
- [CRA] Crane, Aaron. "Linux Ext2fs Undeletion mini-HOWTO".
<http://www.tldp.org/HOWTO/Ext2fs-Undeletion.html> . (March 15, 2004)
- [DCFL] "Imaging and Extracting Solutions"
<http://www.dcfli.gov/DCFL/landE.htm> (March 15, 2004)
- [DMCA] DMCA (Digital Millenium Copyright Act), Oct. 20, 1998.
http://www.eff.org/IP/DMCA/hr2281_dmca_law_19981020_pl105-304.html
(April 16, 2004).
- [ENC] Entries for hash algorithm, cyclical redundancy check, MD5, RAID.
<http://encyclopedia.thefreedictionary.com/> (June 6, 2004)
- [FAM] "Key Features List" (Remote Administrator)
<http://www.famatech.com/radmin/keylist.php> (June 11, 2004)
- [FOL] Free Online Dictionary of Computing. Entry for MAC Address.
<http://foldoc.doc.ic.ac.uk/foldoc/foldoc.cgi?MAC+Address>. (April 14, 2004).
- [FTC] Federal Trade Commission. "Financial Institutions and Customer Data: Complying with the Safeguards Rule".
<http://www.ftc.gov/bcp/online/pubs/buspubs/safeguards.htm> (April 13, 2004).
- [GAT] Gateway 7210 Server Specifications
<http://support.gateway.com/support/manlib/server/7210/7210.shtml>
- [IEEE] "IEEE OUI and Company_id Assignments"
<http://standards.ieee.org/regauth/oui/index.shtml> (March 15, 2004).
- [KES] Kessler, Gary "Linux Magic Numbers".
<http://www.garykessler.net/library/magic.html> (April 13, 2004)

- [MIC1] Microsoft Corporation. "Troubleshooting NTVDM and WOW Startup Errors."
<http://support.microsoft.com/default.aspx?scid=http://support.microsoft.com:80/support/kb/articles/Q196/4/53.ASP&NoWebContent=1>. (April 14, 2004)
- [MIC2] "Microsoft Windows 2000 Server Documentation".
http://www.microsoft.com/windows2000/en/server/help/default.asp?url=/windows2000/en/server/help/SAG_MPmonperf_15.htm (April 16, 2004)
- [MIC3] "Microsoft Security Bulletin MS04-011"
<http://www.microsoft.com/technet/security/bulletin/MS04-011.msp> (June 6, 2004)
- [MIC4] "Indexing Service Event Messages"
http://www.microsoft.com/windows2000/en/advanced/help/default.asp?url=/windows2000/en/advanced/help/ISMain-concepts_78.htm (June 6, 2004)
- [MIC5] "How to Change the Default Installation Paths for FTP and the Web"
<http://support.microsoft.com/default.aspx?scid=kb;en-us;259671> (June 11, 2004).
- [MIC6] "Auditing User Authentication"
<http://support.microsoft.com/default.aspx?scid=kb;en-us;174073> (June 11, 2004)
- [MIC7] "Microsoft Security Bulletin MS04-011 Security Update for Microsoft Windows" Issued: April 13, 2004.
<http://www.microsoft.com/technet/security/bulletin/MS04-011.msp> (June 11, 2004).
- [MIC8] "Microsoft Security Bulletin MS04-012: Cumulative Update for Microsoft RPC/DCOM (828741)" Issued: April 13, 2004
<http://www.microsoft.com/technet/security/bulletin/MS04-012.msp> (June 11, 2004).
- [MIC9] Microsoft Security Bulletin MS04-014: Vulnerability in the Microsoft Jet Database Engine Could Allow Code Execution" Issued: April 13, 2004.
<http://www.microsoft.com/technet/security/bulletin/MS04-014.msp> (June 11, 2004)
- [MIC10] "Critical Update for Windows Media Player (All Versions) for Windows 2000, Windows XP, and Windows Server 2003 (KB828026)"
<http://www.microsoft.com/downloads/details.aspx?FamilyID=af9cf65e-0c55-452e-a0fa-3aa165e667c1&displaylang=en> (June 11, 2004).
- [MIC11] "Microsoft Security Advisor Program: Glossary of Terms."
<http://www.microsoft.com/technet/security/bulletin/glossary.msp> (June 11, 2004)
- [MIC12] "LsaSrv Event 5000 Disables Authentication Packages"
<http://support.microsoft.com/default.aspx?scid=kb;en-us;828873> (June 11, 2004)
- [MKS] Mkssoftware (online documentation and man pages)
<http://www.mkssoftware.com/docs/man1/zipinfo.1.asp> (April 12, 2004).
- [NAC] National Association of College and University Business Officers. (NACUBO) "GLB Act Safeguarding: Compliance with the FTC Safeguarding

Rule Promulgated Under the Gramm-Leach-Bliley Act.”

<http://www.nacubo.org/x325.xml> (April 13, 2004).

- [NIJ] National Institute of Justice. Electronic Crime Scene Investigation: A Guide for First Responders <http://www.ncjrs.org/pdffiles1/nij/187736.pdf> (June 5, 2004).
- [NIST] NIST National Software Reference Library. <http://www.nsl.nist.gov/Downloads.htm> (April 14, 2004),
- [NJ1] Title 2C NJ Code of Criminal Justice 2C:20-2 Consolidation of theft offenses; grading; provisions applicable to theft generally. <http://www.njleg.state.nj.us/> (provides searchable database) (March 15, 2004).
- [NJ2] Title 2A Administration of Civil and Criminal Justice 2A:14-1.1. Damages for injury from unsafe condition of improvement to real property; statute of limitations; exceptions; terms defined <http://www.njleg.state.nj.us/> (provides searchable database) (March 15, 2004).
- [OSI] “OSI 7 layer model tutorial.” http://www.pcsupportadvisor.com/OSI_7_layer_model_page1.htm (June 11, 2004).
- [PAR] Parsons, David “MagicFilter” <http://www.pell.portland.or.us/~orc/Code/magicfilter/> (April 13, 2004)
- [RID] Ridge, Daniel "README" file for bmap ftp://ftp.scyld.com/pub/forensic_computing/bmap/ (March 15, 2004)
- [SKO] Skoudis, Ed and Zeltser, Lenny. Malware: Fighting Malicious Code. Upper Saddle River, New Jersey: Prentice Hall. 2004.
- [US] Gramm-Leach-Bliley Act of 1999. <http://banking.senate.gov/conf/> (March 15, 2004).
- [YUS] Yuschuk, Oleh. Ollydbg. <http://home.t-online.de/home/Ollydbg/> (April 14, 2004).